MSc Artificial Intelligence

Master Thesis

Improving Word Embeddings for Zero-Shot Event Localisation by Combining Relational Knowledge with Distributional Semantics

by

Joop Lowie Pascha

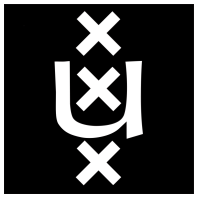
Student-ID: 10090614

November 11, 2018

36 European Credits
January 2018 - June 2018

Supervisor:
Dhr. dr. E. (Efstratios) Gavves
Co-supervisor:
Dhr. N.M.E. (Noureldien) Hussein MSc

Assessor:
Prof. dr. C.G.M. (Cees) Snoek



University of Amsterdam

Abstract

Temporal event localisation of natural language text queries is a novel task in computer vision. Thus far, no consensus has been reached on how to predict the temporal boundaries of action segments precisely. While most attention in literature has been dedicated towards the representation of vision, here we attempt to improve the representation of language for event localisation by applying Graph Convolutions (GraphSAGE) on ConceptNet with distributional node embedding features. We argue that due to the large vocabulary size of language and currently small temporally sentence annotated datasets in scale and size, a high dependency is placed upon zero-shot performance. We hypothesise that our approach leads to more visually centred and structured language embeddings beneficial for this task. To test this, we design a wide-scale zero-shot dataset based on ImageNet to optimise our embeddings on and compare to other language embedding methods. State-of-the-art results are obtained on 5/17 popular intrinsic evaluation benchmarks, but with slightly lower performance on the TACoS dataset. Due to the almost complete overlap in train- and test-set vocabulary, we deem additional testing necessary on a dataset that places more emphasis on word-relatedness; hypernyms, hyponyms and synonyms, which arguably makes language representation learning difficult.

Keywords

Event Localisation · Language Embeddings · Graph Convolutions · TALL-task

Acknowledgements

I would first like to thank my thesis supervisor Dr. Efstratios Gavves at the University of Amsterdam for giving me the freedom and courage to explore the research direction that ultimately lead to the work presented here. In addition, I want to thank my co-supervisor, Noureldien Hussein, for sharing his initial views upon his proposed research topic of *Temporal Localisation of Activities in Videos given Natural Language Text* that helped me shape my view of the problem, while giving me the flexibility to come up with my own approach and contribute to the research field.

Throughout the years I have also had many teachers, friends and family members that put me in the position I am in today. My heart goes out to all the people who had a positive impact in my life and provided me with valuable life-lessons.

Special mention goes towards my mother, Ana, for being there for me through the good and bad times. Without you I would not have been in the situation I am in today, and as such I dedicate this thesis to you.

Lastly, I want to thank the friends I have come to know and appreciate throughout the long working days during the Master Artificial Intelligence. It is these interactions I have valued above all others, both intellectually and emotionally, and made the long working hours worth it: Marco Federici, Janosch Haber and Muriel Hol.

Joop Pascha

November 2018

Contents

1 Introduction *7*
1.1 Improving Language Embeddings with Event Localisation in Mind 7
1.2 Language and the Relation to the Physical World 11
1.3 Our Approach 13
1.4 Research Questions 14
1.5 Hypotheses 15
1.6 Contributions 15
1.7 Outline 16
1.8 Experiments and Relation to the Research Questions 16

2 Background *18*
2.1 The TALL Task 18
2.2 Video Representation - Modeling Difficulties 20
2.2.1 Spatio-Temporal Video Representation 20
2.2.2 Datasets 23
2.2.3 Computational Efficiency 28
2.3 Language Representation - Modeling Difficulties 30
2.4 Remarks & Sub-Conclusions 33
2.5 Graph Convolutions & Language Embeddings 33
2.5.1 Retrofitting vs Graph Convolutions 34

3 Related Work *37*
3.1 TALL Model Architecture 37
3.1.1 Modality Feature Extraction 38
3.1.2 Sampling Training Examples 38
3.1.3 Loss Functions 39
3.1.4 Evaluation Setup 39
3.1.5 Observed Difficulties 40
3.2 The Addition of Language in Action Localisation 40
3.2.1 From Words to Word-Embeddings . 41
3.2.2 Intrinsic Evaluation Methods 42
3.3 GraphSAGE 43
3.3.1 Training 46
3.3.2 Aggregators 46
3.3.3 Sampling 47
3.4 Zero-shot Learning 47
3.4.1 Zero-Shot Evaluation Metrics 49

4 Methods *50*
4.1 Problem Formulation 50
4.2 Overview of Experiments 50
4.3 I - Zero-shot Cross-Modal Embedding Space Evaluation 54
4.3.1 Objective & Relations to Research Questions 54
4.3.2 Methods Overview 55
4.4 II - GraphSAGE-ConceptNet Embeddings 57
4.4.1 Objective & Relations to Research Questions 57
4.4.2 Methods Overview 57
4.5 III - TALL with Sentence Embedding Replacements 59
4.5.1 Objective & Relations to Research Questions 59
4.5.2 Methods Overview 59

5 Experimental Setup *61*
5.1 I - Zero-Shot Evaluation of Cross-Modal Embedding Space 61
5.1.1 Dataset Creation Details 61
5.1.2 Cross-modal Embedding Baseline . 63
5.1.3 Training Objective & Evaluation Benchmark 67
5.1.4 Architecture Selection & Training . . 68
5.1.5 Qualitative Analysis using Flickr30k 74
5.2 II - GraphSAGE-Conceptnet Embeddings 75
5.2.1 ConceptNet Analysis & Graph Comparisons 75
5.2.2 OOV Matching & Dataset Creation . 77
5.2.3 GraphSAGE Training & Parameter Selection 80
5.3 III - TALL with Embedding Sentence Replacements 83
5.3.1 Averaging: from Word to Sentence Embeddings 84
5.3.2 Infersent: from Word to Sentence Embeddings 85
5.3.3 TALL Training & Reproduction . . . 86
5.3.4 TACoS Analysis for Zero-Shot Test-Set 87
5.3.5 Charades-STA Analysis for Zero-Shot Test-Set 87

5.3.6 One-hot Encoding of Words Alternative . . . 87

6 Results & Analysis . . . 89
6.1 I - Zero-shot Results of Cross-Modal Embedding Space . . . 89
6.2 II - GraphSAGE-ConceptNet Embeddings Results . . . 91
6.2.1 Quantitative - Intrinsic Evaluation . 91
6.2.2 Qualitative Results - TSNe . . . 92
6.2.3 Flickr30k Zero-Shot Evaluation Analysis . . . 92
6.3 III - TALL with Sentence Embedding Replacements . . . 95

7 Discussion . . . 97
7.1 Restating Hypothesis . . . 97
7.2 Relations between Results & RQs . . . 98
7.2.1 RQ1 - Improved Alignment? . . . 98
7.2.2 RQ1 - Remarks about Methodology 99
7.2.3 RQ2 - Zero-shot Dataset? . . . 100
7.2.4 RQ2 - Remarks about Methodology 101
7.2.5 RQ3 - TALL-task Performance? . . . 102
7.2.6 RQ3 - Remarks about Methodology 104

8 Conclusion . . . 105
8.1 Future Work . . . 106

Figures . . . 114

Tables . . . 120

Acronyms . . . 121

Appendices . . . 122
A Intrinsic Evaluation Methods Tables . 122
A.1 Aggregator Function vs Feature Initialization . . . 122
A.2 Random vs Non-Random Path . . . 122
A.3 Hops Length vs Aggregate Function 122
A.4 Random Walks Count vs Agregator Function . . . 123
A.5 Dropout vs Aggregate Function . . . 123
B Others . . . 123
B.1 Sentences used for TSNe Visualization 123
B.2 Word-Embeddings and References . 123
B.3 Cosine similarity Numberbatch and Our embeddings. . . . 124

1 Introduction

Forty-eight hours of videos are uploaded to Youtube every minute with future-projections only indicating that the amount of video footage created by consumers and companies increases[1,2]. For many applications, including the search and recommendation of content, it is necessary to understand what occurs within this content. However, manually labelling and transcribing these videos is for humans a time-intensive task with limited possibilities towards speeding up this process. Therefore, there is a high demand for methods that can automatically search, annotate and recommend these videos for the efficient retrieval of information. Any solution towards this general problem description places a heavy reliance on how visual cues (e.g. *objects*) correspond to their linguistic counter-part (*words*). As of yet, no consensus has been reached towards how such a suitable cross-modal embedding space can be obtained that allows for matching textual descriptions with video segments of variable size[3,4]. Although recently this particular research domain has gained increased traction within the field of Computer Vision.

[1] Fu et al. (2014)
[2] Caba Heilbron et al. (2015)
[3] Nguyen et al. (2017)
[4] Xu et al. (2017)

1.1 Improving Language Embeddings with Event Localisation in Mind

In this work, an attempt is made to improve upon the representation of language specifically for the task of event-localisation in videos given natural language text. Gao et al. (2017) recently introduced a novel challenge called the Temporal Activity Localization via Language (TALL)-task in which the objective is to localise *any* textual description in natural language text within videos. Whereas current event-localisation approaches attempt to localise only a small number of event-"classes" in videos within a narrow domain, Gao et al. instead use natural language text to represent a variety of events using word embeddings. The use of natural language changes the approach from a relatively simple classification task to a regression problem in which significantly more emphasis is placed upon the representation of language. Therefore in our work, we specifically focus on improving this representation of language for event-localisation in videos.

To improve the representation of language for the particular task of event-localisation, first hypotheses were formulated about which properties of language embeddings were deemed most relevant for matching textual events with vision (meaning: ≈feature representations of images). Subsequently, a novel approach was designed to create language embeddings based

on these properties. In an attempt to isolate whether these properties indeed lead to the hypothesised improved task-performance in event-localisation, additional experiments are introduced to obtain a quantitative score to the extent these properties were apparent in a multitude of different language embeddings. Then we compare these scores for a variety of embeddings, including our own, with their actual downstream task performance to test whether these properties indeed lead to improved performance. The remainder of this chapter is intended to provide the reader with the additional background needed to understand what lead to the formulation of our approach, including; the difficulties of this particular task, our most essential realisations that helped shape our hypotheses and end with the research questions that we attempt to answer in this work.

Current methods in video understanding mainly focus on the representation of vision and simplify the representation of language to only a select number of pre-defined classes[5]. Arguably, this task-design could be improved upon by allowing natural language text to be used to not be limited to only these select number of event categories. However, in order to go from only a select number of target classes to using natural language text, numerous challenges arise. Whereas in the former task design a complete overlap between the training- and test-set class-categories exist, this is not the case when natural language is used to represent events. With only limited visual-textual correspondences during training-time and the vast vocabulary size of natural language text, it becomes essential to relate the seen vocabulary during training to the unseen vocabulary during test-time. This arguably makes this problem close to a Generalised Zero-Shot Learning (GZSL) problem-setting in which high performance is vital on both seen and unseen vocabulary during the test-time. The objective, therefore, becomes to transfer knowledge from the training- to test-setting. De Boer et al. (2017) in their work focus on the semantic reasoning in zero example video event retrieval which is close to the former problem description. De Boer et al. describe the absence of appropriate datasets as the primary challenge with two properties of concepts mainly contributing to this; the *level of complexity* and *level of granularity* concepts can be described at.

[5] Gao et al. (2017)

The complexity of a concept refers to whether an event is described on the *low*-level of objects, *mid*-level of basic actions or *high*-level of complex sequences of movements. For example on the object-level a description could be; *Humans are kicking a ball and try to score in each others goal*, on the mid-level; *People try to outscore each other through passing and shooting*, and high-level; *they play football*. The granularity on the other hand, states that *Chihuahua* is a more specific example of a *dog*. With the

English vocabulary containing at least 171476 unique words[6] and the event-localisation datasets being relatively small which span only a small subset of these words (e.g. [7,8]), this arguably places significant emphasis on how concepts or words in the training-set *relate* to any word in the English vocabulary.

Another difficulty that arises when natural language text is being used to represent language instead of a select few action categories is the added uncertainty that comes when matching vision with language. Whereas in event classification the classes are assumed to be non-overlapping and binary, with natural language text calculating the similarity between word-vision correspondences is more difficult due to the increased subjectivity. However, the added benefits of using natural language text are that it can be potentially used to localise *any* event that can be described in natural language text and also less emphasis is placed upon on artificially created datasets that consist of subjectively created *event-classes* that require many training examples per class.

In the work of Gao et al. (2017) that formally introduced the TALL-task, the most emphasis was placed upon obtaining a suitable model to learn a cross-modal embedding space in which language can be accurately matched with parts of a video. The two modalities, represented by language and vision respectively, are represented by general purpose language embeddings obtained using Distributed Semantic Models (DSMs), e.g. *word2vec* or *Skip-Thought*, and visual features extracted from pre-trained Convolutional Neural Networks (CNNs) respectively. We refer to the space in which both modalities can be matched as the *cross-modal embedding space* which is learned by a parameterised model. In this semantic space, the distance between both feature-representations should ideally reflect how semantically similar both representations are.

To improve the representation of language and facilitate the alignment between vision and language for this cross-modal embedding space, a novel approach is designed here that relies upon a Graph Convolutional Network (GCN) being applied on a Knowledge Base (KB) combined with DSM node feature-representations. Due to the large vocabulary size of language and limited visual-language correspondences in current training-sets, the problem is formulated as a missing-data problem in which there is a heavy dependency placed upon GZSL. To perform well in this task-setting, high performance is necessary on both seen classes during training and unseen classes during testing. Which in our problem-setting loosely corresponds with (un)seen words and visual examples.

In order to accurately match vision with text for event-localisation in videos, we hypothesise that more structured language embeddings are required than current DSMs provide. The additional structure could enhance the knowledge trans-

[6] source from oxforddictionaries

[7] Regneri et al. (2013)

[8] Sigurdsson et al. (2016)

fer from seen to unseen words in the cross-modal embedding space. Also within the specific domain of event-localisation or recognition, language embeddings that more prominently feature visually grounded relations between the words in the vocabulary were deemed beneficial for subsequent matching with vision. Leading up to our approach, first a literature study was conducted to identify the most recent developments and encountered problems within the domain of video-understanding. Second, based on this literature overview, a literature-gap was identified leading up to our approach. The two main problems that were identified using DSMs for event-localisation given natural language text are now described in more detail.

First, DSMs approaches[9] learn the relations between words using the *distributional hypothesis* on large text corpora coming from a different domain than the visually centred textual descriptions as can be seen in event-localisation. Arguably this leads to sub-optimal word-representations for this task. The *distributional hypothesis* states that the linguistic items used in similar context have a similar meaning. Considering the data sources that these models are trained on, e.g. Wikipedia, the resulting embeddings are expected to encapsulate relationships that are more centred around the historical context of events in contrast to the more visually grounded relationships used to describe events in videos. For example, to match the textual query *a spinning top on the table* it is useful to have language embeddings that more prominently features the functional relationship that a *top* can *spin* in order to match it with the visual motion of *spinning*. In contrast, current DSM approaches rely on data sources such as Wikipedia which focus more on historical context of entities and objects, e.g. *Barack Hussein Obama II, is an American politician who served as the 44th President of the United States*[10]. KBs such as ConceptNet, however, are centred around objects and their functionality, which could potentially be used as an alternative method towards obtaining language embeddings that more prominently feature visually centred relations in them.

Second, to allow the accurate matching of *any* linguistic description with vision arguably a significant dependency is placed upon how one can relate the known vocabulary during training to the unknown vocabulary during testing. This arguably makes this problem closely related to the GZSL task-setting. In contrast to Zero-Shot Learning (ZSL) which only considers the model's performance on unseen classes during testing, in GZSL approaches also the performance on the classes already seen during training is taken into account. For our purpose, these (unseen) classes loosely correspond with the vocabulary on the language-side with matching visual representations on the vision-side. As the same image or video can be described in an almost infinite number of ways using dif-

[9] e.g. Word2Vec (Mikolov et al. (2013a)), GloVe (Pennington et al. (2014)), LexVec (Salle et al. (2016))

[10] Sentence taken from the Wikipedia page of Barack Obama https://en.wikipedia.org/wiki/Barack_Obama

ferent words with emphasis on different visual cues and level of complexity, the amount of visual-textual correspondences during training time can always be considered only a small fraction of the total amount of possible descriptions. Therefore, a large part of the challenge of finding a cross-modal embedding space can be considered an alignment issue in which the relationships learned during training-time between the vision and language modality need to be able to generalise to a high extent to unseen visual-imagery and textual-descriptions during testing. For this reason, an attempt was made towards optimising language embeddings specifically for better zero-shot performance when used in a cross-modal embedding space setting with vision. The structure of a KB in contrast to DSM approaches was expected to lead to more structured language embeddings and therefore better suited to transfer knowledge to unseen words in the cross-modal embedding space.

Inherently, in order to match language and vision in a cross-modal embedding-space, it can be beneficial to understand how we as humans use language to describe the visual world. In the following section, this is explored and further illustrates how the usage of a KB can help towards revolving the aforementioned two problems.

1.2 *Language and the Relation to the Physical World*

The question can be posed, what exactly is the relationship between language and vision? Arguably, language has evolved to describe our surroundings in a simplified abstract way that allowed people to effectively communicate ideas and refer to the same visual surroundings in the real world. Therefore, there must exist a commonly shared latent representation between people; an abstraction of the physical world, that is tapped into by communicating through language. In this condensed representation, some objects or events are intuitively closer to us. For example we find that *women* and *child* are more similar or closer to each other than *child* and *make-up*.

A possible explanation for our intuition that some concepts are closer together than others is the clustering of these objects in the real physical world. To encode the world in a latent abstract representation with limited capacity, a possible analogy of the storage device which is our brain, clustering of visual representations could potentially be an efficient and practical solution. As *make-up* is visually more frequently seen near *females*, while for example a *banana* is not, it also makes sense to cluster these concept abstractions closer together as they are observed in a similar context. Many intrinsic evaluation benchmarks have been designed to try to capture this perceived

similarity between words by humans, which subsequently have been used to test whether the language embeddings trained by parameterised models exhibit the same similarities between words as a measurement of their quality. Popular similarity based tasks include MEN[11], MTURK[12] and WS[13].

In particular, the highly structural formation of words that make up language, with it is many; hypernyms, hyponyms, antonyms, synonyms and other types of relations, could provide a peek into the underlying structure the outside world is encoded in within our brain. This representation could allow for more efficient encoding of concepts, such that concepts that visually appear closer together in the real world also appear closer together within this language hierarchy. For example a *cat* is a *pet*, and a *pet* is owned by a *human*, could possibly be seen as an indication that the concepts of *humans* and *cat* also co-occur closely together in the physical world.

So how can this view upon language as a way to describe a latent representation of a hierarchical abstraction of the world help in modelling textual and visual similarity for the task of event-localisation in videos? Given that this hierarchy is known, this eases the subsequent matching of vision and language when only limited visual-and-textual correspondences are available during training time. This hierarchy allows for transferring knowledge from the known concepts during training-time to previously unseen concepts in test-time. Of course, this hierarchy is unknown in practice, but arguably knowledge bases such as WordNet and ConceptNet already attempt to make these semantic clusters in language concrete by introducing a set of ternary relations <subject, relation, object>. An example of a sub-graph of these relationships in ConceptNet is shown in Figure 1.1.

[11] Bruni et al. (2014)
[12] Radinsky et al. (2011)
[13] Agirre et al. (2009)

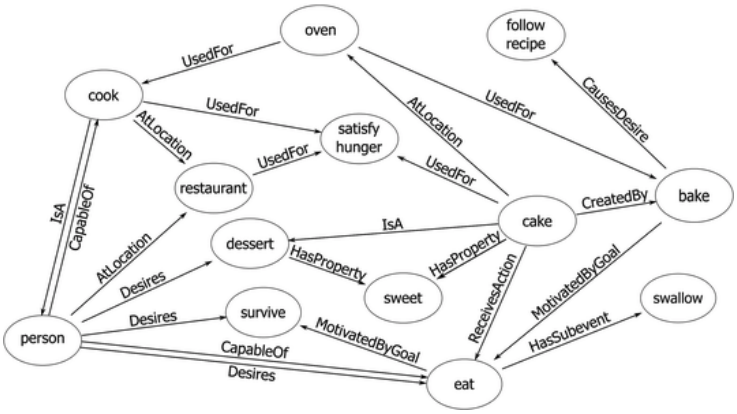


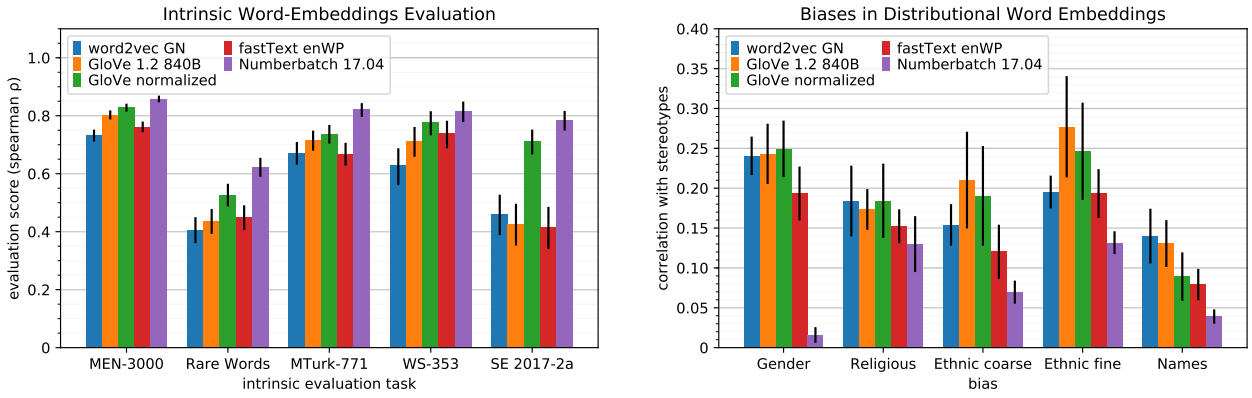
Figure 1.1: ConceptNet subgraph. Relations between concepts are shown by arrows and are directional. The text above or below the arrows demonstrate the relationship type (e.g. UsedFor, AtLocation). Relational data from ConceptNet as shown here can potentially be combined with semantic node embedding features to obtain better language embeddings for event-localisation. Figure reproduced from Speer and Havasi (2013).

The hierarchy that is contained within these KBs could potentially be harnessed as an alternative way to obtain language embeddings. Whereas in DSM approaches there is limited control over which relationships are being learned resulting

in general purpose word-embeddings, using only a selection of a KB like ConceptNet could potentially allow for more control over which relationships are being learned to specifically gather towards a task of interest. For example, one could only select specific relationships that are deemed useful for event-localisation purposes. In Figure 1.1 an example is shown of the relations between concepts in ConceptNet that could be used for such a selection.

Another limitation of DSM approaches is that they rely on large quantities of data in which words that are more frequently appearing in the same context are considered more semantically similar. However, intuitively repeating the same sentence does not make the words within the sentence more similar to us. Nonetheless, DSM approaches are due to its distributional hypothesis vulnerable to this frequency bias. An implication of this is that for example a *man* is more associated with the word *boss* while *female* is more associated with the word *cooking*, an undesirable property for many practical applications[14]. By using relational knowledge from KBs to obtain these embeddings, the frequency in which these relationships appear in written-text can be partially neglected (explain in more detail in Section 1.3). However, it is important to note that even the hierarchical structure of KBs are still subjected to our own biases and therefore not completely without biases.

[14] Speer (2017)



(a) Word embedding evaluation comparisons. Higher is better.

(b) Word embedding bias comparisons. Lower is better.

Figure 1.2: Comparison between popular language embedding methods on intrinsic evaluation benchmarks and bias-metrics. Figure reproduced from here. In intrinsic evaluation tasks the similarity between word-pairs is calculated based on human judgment and is then compared to the similarity these word-pairs have in the language-embeddings as a measurement for success.

1.3 *Our Approach*

Speer and Lowry-Duda (2017) recently showed the success of combining relational knowledge found in knowledge bases such as ConceptNet, with distributional word embeddings to obtain improved word embeddings using a technique called retrofitting[15]. In their work specific focus was dedicated to decreasing the effect of a variety of biases that are apparent in DSM while improving the state-of-the-art (SOTA) in many of the intrinsic evaluation benchmarks. This resulted in language embeddings called *Numberbatch* of which the results can be seen in Figure 1.2 which shows promising signs that relational knowledge from ConceptNet can be used to improve

[15] Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2014). Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*

the quality of general purpose language embeddings.

Recently also a new type of algorithms have been introduced that could combine relational knowledge and distributional semantics; GCNs. In specific, the work of Hamilton et al. (2017) that introduced the Graph Convolutions (GC)-based method *GraphSAGE*, that allows learning node-embedding feature representations for each node in large-scale graphs in an unsupervised fashion and in an inductive setting (for a more in-depth explanation see Section 3.3). The representation of each node is dependent upon the node's local neighbourhood while distant nodes are enforced to be dissimilar. In addition, Hamilton et al. propose a variety of learn-able aggregator functions to combine this local neighbourhood information of each node while allowing each node in the graph to have an additional n-dimensional feature vector. To the best of our knowledge, this method has not been applied on a KB yet in an attempt to obtain language embeddings.

In this work, we further explore the possibility of combining relational knowledge with distributional semantics specifically to obtain improved language representations for event localisation given natural language text in videos. By applying the recently popularised neural network architecture; GCNs on a specific sub-selection of ConceptNet to which node-feature representations are added from popular DSM approaches, it is expected that more structured language embeddings are obtained. This additional structure is hypothesised to result in improved generalisation in a GZSL task-setting when compared to current language embeddings obtained using DSM approaches. In addition, the object centred focus of ConceptNet in which many of the relationships are visually grounded is expected to improve the alignment with visual-features to obtain a cross-modal embedding space.

This new approach towards obtaining language representation adds additional challenges including; the question whether Graph Convolutions can successfully be applied to the domain of ConceptNet under our task-settings, how to correct the mismatch between the vocabulary of ConceptNet and DSM methods to add appropriate node-embedding features, and how to evaluate whether the resulting language embeddings (1) contain the desired properties we hypothesized and (2) whether this lead to actual task improvements.

1.4 *Research Questions*

The following research questions were specifically attempted to be answered in this work;

- **RQ1** Are language embeddings that are obtained by combining both distributional semantics and relational knowledge, better able to be aligned with the visual-features

for zero-shot purposes than using distributional semantics alone?

- **RQ2** How can a wide-scale GZSL evaluation dataset be designed that covers the broad nature of events in videos and allows comparing different language embedding methods in their ability to be matched with visual-features given a GZSL task-setting?

- **RQ3** Is a higher zero-shot performance on the evaluation dataset obtained in RQ2 actually indicative of increased task performance in event-localisation in videos given natural language text?

1.5 *Hypotheses*

- **H1** For event-localisation in a video given natural language text the large vocabulary size of language with the many hypernyms, synonyms and other relationships between individual words, require more structured language embeddings than current DSM provide in order to improve the transfer of knowledge from seen to unseen vocabulary similar to a GZSL task setting.

- **H2** For the matching between visual-features and language-embeddings in a cross-modal embedding setting, it is beneficial if the relationships that the language embeddings entail are more visually grounded.

1.6 *Contributions*

The contributions of this work include the following;

1. To the best of our knowledge, we are the first to apply graph convolutions on ConceptNet with node feature representations taken from distributional word-embedding approaches as an alternative way to obtain language embeddings.

2. The obtained language representation show competitive results on 14 of the 17 used intrinsic evaluation methods while reaching SOTA in the metrics AP, BLESS, ESSLI_1A, ESSLI_2C and RW.

3. Our language embeddings obtain similar but slightly lower performance than the current SOTA in the TALL-task[16] substituting only the language representation the approach of Gao et al. (2017). Further inspection showed that for this task still high performance could be obtained when words were represented using a 1-hot encoding rather than lower dimensional word-embeddings. This demonstrates that for this task there is limited reliance upon the transfer of knowledge between the train- and test-set vocabulary. We argue that due to the limited visual-textual correspondences in

[16] Gao et al. (2017)

current event-localisation datasets and the nature of this problem, this is not a realistic evaluation-setting. Therefore for the evaluation of this task, we deem the introduction of a novel dataset necessary which places more emphasis on the transfer of knowledge by containing more vocabulary variety and less overlap between the train- and test-set vocabulary.

1.7 Outline

The remainder of this work is broken up in the following sections. In the Background Section (2) the TALL-task is introduced, an overview is provided towards the problems that are discussed in literature when learning a language and visual representation and lastly a comparison is made between GCs and retrofitting as a technique to combine relational knowledge with distributional semantics. This chapter is intended to give the reader a basic understanding of the topic. In Related Work (3) a detailed explanation is given of the used TALL- and GraphSAGE model-architecture that was used to obtain our results as well as the evaluation methods of word-embeddings and zero-shot learning approaches. Thereafter in Methods (4) the TALL-task is formalised, and an overview is provided into the three experiments conducted in this work in an attempt to answer the research questions posed in the Introduction (1.4). In Experimental setup (5), the challenges that were faced are addressed. Including the creation of a suitable zero-shot dataset to evaluate to which extend language embeddings are suited to be aligned with visual-features in a GZSL setting (5.1.1), the challenges faced when obtaining language embeddings using our given approach (5.2) and how to go from word-embeddings to the down-stream task performance (5.3). An overview of the Results (6) is then provided after which a Discussion (7) follows that questions or give strength to our methodology were appropriate. Lastly, we Conclude (8) with a summarisation of our key findings and recommendations for future work.

1.8 Experiments and Relation to the Research Questions

RQ1 is attempted to be answered in Experiment III (4.5) where we compare our obtained language embeddings in the evaluation setup as formulated by Gao et al. (2017) on the TALL-task to test whether our hypothesis (1.5) indeed resulted in improved task-performance. As in our hypothesis we argued that for the TALL-task the transfer of knowledge from seen to unseen vocabulary is important and more visually grounded language embeddings are beneficial, we conduct two small experiments to test whether this is indeed true. To test the former,

in Section 5.3.6 we replace the vocabulary to a 1-hot encoding of words to minimise the reliance upon knowledge transfer and measure the performance on the TALL-task. As the TALL-task was formulated as a direct response to the critique that current methods represent language as a simple 1-hot encoding of only a select few *event*-classes, a suitable task-evaluation method would place emphasis upon word-*relatedness* rather than the direct matching of classes (or words). Therefore representing a 1-hot encoding of words was expected to result in a relatively low performance given a suitable task-evaluation setup.

Next, to test whether our embeddings featured relationships that had visual correspondences, in Section 5.1.5 we conduct an experiment using the Flickr30k dataset. By POS-tagging the sentences in the Flickr30k dataset and ranking the word-image similarity scores of dissimilar and similar pairs given a trained cross-modal embedding space, it was expected that a more qualitative comparison could be made between language embeddings and their ability to be aligned with visual-features (5.1.5). However, as the obtained cross-modal embedding space as obtained in Experiment II (4.4) was unable to accurately match word-image pairs on the Flickr30k dataset, any further analysis was not considered meaningful. Therefore, this question remains unanswered and only for completeness this experiment is shortly discussed.

For RQ2 we design Experiment II (4.4) where we explore whether the hierarchical structure of ImageNet can be used to create a zero-shot evaluation benchmark with the objective of testing *general* zero-shot performance. Finally to answer RQ3 and observe whether increased zero-shot performance also leads to increased TALL-task performance we compare the results we obtained using the dataset obtained in RQ2 with the task-performance obtained during answering RQ.

2 Background

Prior to the formulation of the research direction as mentioned in the Introduction (1), an extensive literature study was conducted in an attempt to provide an overview of the identified problems within the domain of event-recognition and localisation. Based on these findings a literature-gap was identified that revolved around obtaining an improved representation of language as current literature greatly simplifies the use of language to a simple classification task or rely on general purpose word-embeddings obtained by DSM methods. The findings from this literature study and the preliminary background needed to understand our approach, are discussed in this chapter.

First a more formal introduction of the TALL-task is given in Section 2.1. Subsequently in Section 2.2 three major difficulties deemed most important for obtaining an accurate representation of vision in video-understanding are discussed. Thereafter in Section 2.3 a more in-depth overview is provided towards the challenges revolved around obtaining an accurate representation of language for event-localisation or action classification. Our final remarks and sub-conclusions regarding these findings are then summarised in Section 2.4 which were used as a starting point for our approach. Lastly, as our approach shares similarities between the retrofitting technique that was used to obtain the Numberbatch word-embeddings leading to the current SOTA in language embeddings, Section 2.5 is used to point out the most important differences.

2.1 The TALL Task

Gao et al. (2017) formalise the challenge of finding exact temporal boundaries in untrimmed videos of free-form text queries as the TALL-task. In contrast to trimmed videos which are used for video-classification (also called event recognition) with only one target label per video, in untrimmed videos only a segment of the video corresponds to the textual description of the event. Event-localisation can, therefore, be seen as a task where first it is required to find *where* an event occurs after which there has to be determined *what* it is about. Different from the traditional action localisation task, the TALL-task describes *what* it is about in natural language text without any self-imposed structure (*free-form*) instead of a select list of pre-defined actions or events, therefore putting increased emphasis on the representation of language. From now on the terms *events* and *actions* will be used interchangeably to refer to a textual description with clear visual correspondences that can be localised in videos.

In a video multiple and sometimes even overlapping events can occur with a significant part of the video not being specifically gathered towards any particular action. Therefore one of the most challenging tasks of event-localisation in untrimmed videos is being able to separate the *most* salient part of an event from the rest of the video[1]. Whereas in a classification task there is relatively little ambiguity about the correct class from the select and pre-defined list of classes, the exact temporal boundaries of an event are often subjective and therefore debatable. With the introduction of the TALL-task the field of Computer Vision (CV) has progressed from the classification of objects to entire videos, and the localisation of actions in untrimmed videos given a pre-defined list of action-*classes* to also freeing up on this final constraint. Arguably, this brings us one step closer for these methods to be applied in real-world applications that require a search in videos in our own *natural* language.

[1] Nguyen et al. (2017)

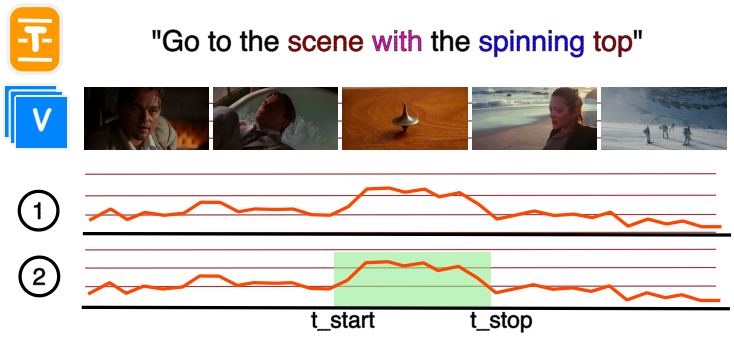


Figure 2.1: The aim in the TALL-task is to find the temporal boundaries of an event described by a textual description T in video V. (1) A cross-modal embedding space is learned that should give high activation for corresponding V and T. (2) Thereafter a segment proposal network is trained that learns based on the activation output of step (1) to predict the temporal boundaries *t_start* and *t_stop* of the event described by T.

Gao et al. (2017) subdivide the TALL-task in two separate steps. First, (1) the design of a text and video representation that allows for creating a joint-representation of language and vision in which the similarity of the two can be measured. We refer to this as the *cross-modal embedding space*. Second, (2) the ability to accurately locate the actions using the similarity scores obtained from the cross-modal embedding space using sliding window based approaches of limited granularity to account for actions of variable length. How these tasks are dependent upon each other is shown in Figure 2.1. Inherently, (2) is dependent upon the feature representation obtained in (1) which therefore propagates potential sub-optimal language or vision representations further down the model. Gao et al. focus mostly on obtaining an appropriate model for (2) while simplifying the language and vision representation by extracting features from models trained separately, *Skip-Thought* and *Inception-V1*, on different tasks to base their cross-modal embedding on (1). Therefore first an effort was made towards providing a literature overview of the current difficulties in finding an appropriate video and language representation.

2.2 *Video Representation - Modeling Difficulties*

Finding a suitable representation of vision is frequently brought up as one of the most challenging tasks for accurate action localisation. Concerns in literature are frequently described as the lack of (1) suitable spatiotemporal video representation for accurately capturing the large intra-video variation, (2) suitable datasets for this task and (3) the computational efficiency in which this is obtained (Figure 2.2). These are now further discussed.

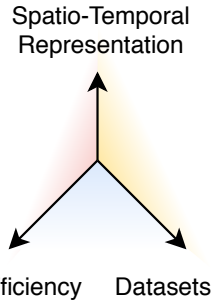


Figure 2.2: A simplified overview of three identified problem-areas of event-localisation in literature. Current approaches can be roughly divided along these three dimensions; *Upper center*: finding a suitable visual representation. *Bottom left*: the computational efficiency in which the localisation and classification of action occurs. *Bottom right*: finding datasets suitable for event-localisation in both size and variety to accomplish this task. The axis are to a certain extend dependend upon each other.

2.2.1 *Spatio-Temporal Video Representation*

Dai et al. (2017) describe that although the localisation of objects in images has been widely studied, localisation of activities in videos has received less attention. The primary reasons Dai et al. accredit to this are the increased computational cost associated with working within the video domain combined with the lack of large annotated datasets. Yuan et al. (2016) specifically mention the difficulty of representing time to allow to capture events of arbitrary length. Yang et al. (2018) describe that the task of how to accurately perform temporal action localisation is still an open question, while Nguyen et al. (2017) argue that the lack of appropriate methods to obtain suitable video representations is the main challenge in action localisation.

Xu et al. (2017) stress the necessity of extracting meaningful spatiotemporal features to accurately localise the start and end times of each activity. Current approaches have the drawbacks that they do not learn deep representations in an end-to-end fashion, but instead rely on hand-crafted features or deep features extracted from CNNs trained on a different task. Xu et al. argue that these off-the-shelf representations may not be optimal for action localisation because of the tremendous diversity of videos. The vast diversity seen in videos according to Caba Heilbron et al. (2016) comes from the considerable variation in motion, scenes, and objects involved, styles of execution, camera viewpoints, camera motion, background clutter and occlusions. This makes learning general discriminative features for videos for a wide variety of domains difficult.

What makes videos different from images is the large spatiotemporal correlation found between consecutive frames. These correlations can be captured by complex motion features for the tasks of the localisation or classification of actions, either by using CNNs trained on videos using 3D filters or static approaches. Static approaches, such as $TV - L^1$ and dense-trajectory estimation, in contrast to CNN based approaches, are not optimised on a specific dataset and target domain. Instead, they focus on separating object motion from camera

motion estimation using for example homography estimation[2], the tracking of SURF-descriptors across frames or variational methods to obtain optical flow estimations.

C3D and the more recently introduced I3D are popular CNN based methods to encapsulate motion and visual features in one joint representation. In these approaches, a visual feature map is generated containing a representation of multiple frames at once. As these methods take up somewhere between 0.4 to 2.56 seconds of visual input at a time[3], finding an accurate frame-level prediction of event boundaries for action localisation is therefore difficult using these approaches. Potential solutions for this particular problem have been proposed by for example Yang et al. (2017) that use Convolutional-Deconvolutional-Convolutional filters to allow for more accurate frame-level predictions.

The use of 3D filters in methods such as *C3D* and *I3D* come at the cost of increased model complexity which increases the risk of over-fitting. Recently, Carreira and Zisserman (2017) introduced their SOTA I3D model on video classification benchmarks which is based on the older Inception-V1 architecture that focused on computational efficiency allowing for increased depth and width of the model's architecture. By inflating the 2D filters pre-trained on Imagenet into the temporal dimension as a smart initialisation method, improved training-time, and classification accuracy was obtained due to decreased over-fitting possibilities.

Xie et al. (2017) further improved upon the I3D architecture by using temporally separable convolutions, introduced spatiotemporal gating mechanisms with additional spatial- and temporal-pooling. By shifting the temporal depth from the bottom-layers to the top-layers, the required number of parameters were reduced and the performance increased. Despite these efforts towards reducing model complexity, both models still were trained using respectively 64 and 56 GPUs with synchronous stochastic gradient descent on the largest video classification dataset at the time; Kinetics, in order to combat over-fitting. Training on a large dataset can be seen as an additional form of regularisation by decreasing the influence of each data-point. While the large amount of GPUs is necessary to compensate for the increased memory requirements of one training example due to the temporal dimension (n-consecutive input frames) and increased model size when compared to CNNs that take in only a single image (3D filters).

For many video-related tasks, e.g. the localisation and classification of actions, features are extracted from SOTA video classification models. The architecture of these models can be roughly divided into five popular model architectures as displayed in Figure 2.5. Depending on the architecture type, inputs of the model either allow the use of one or multiple

[2] Wang and Schmid (2013)

[3] Carreira and Zisserman (2017)

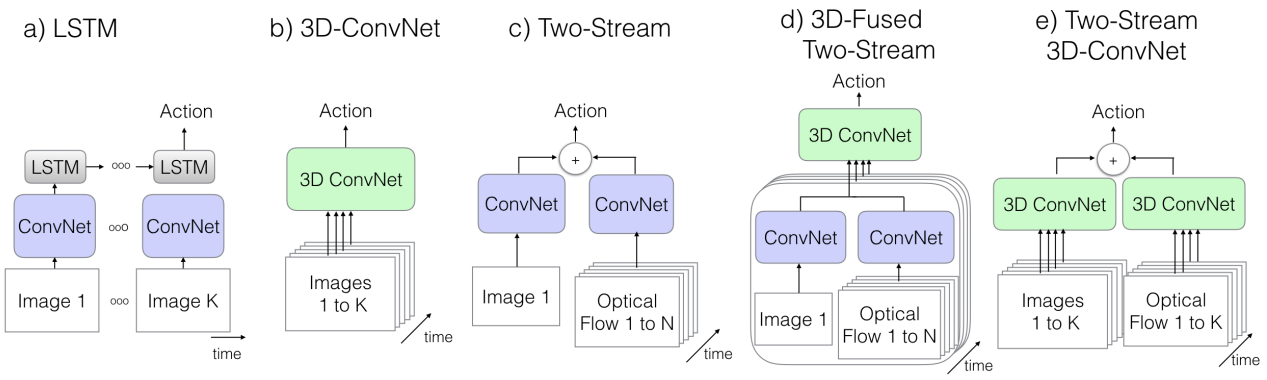


Figure 2.3: Popular architectures for learning visual video representations that include motion. The models input differ in their representation of time, e.g. on the frame-level (a,c,d) vs. multiple frames (b,e), without (a,b) or with additional motion information (c,d,e). The model architectures also differ in the moment motion information is aggregated, e.g. late (c) vs early fusion (a). Motion patterns can be learned using 3D filters (b,d,e) or 2D approaches (a,c). Figure reproduced from Carreira and Zisserman (2017).

images with or without the addition of optical flow estimations between these frames. Optical flow takes the displacements of intensity patterns into account and therefore can be seen as a background masker in which the moving parts of the image are separated from the stationary background. Also, frequently an attempt is made to distinguish camera motion from object motion, which is especially useful to detect motion patterns. Popular optical flow estimation methods are TV-L1[4,5,6] and dense flow optimization[7], with a great overview of the different methods provided by Fortun et al. (2015).

Another dimension of differences between network architectures used in classification tasks is when the two streams of information, optical flow and RGB images, are fused. The most popular fusion techniques are called late and early fusion, of which the differences are in more detail described by Karpathy et al. (2014). In early fusion, the input streams are brought together in the original feature space while in late fusion the input streams both modalities are fused in semantic space. The same trade-off between early and late fusion is frequently made within the spatiotemporal space of models. Here temporal information is frequently traded for spatial depth further down the network. Although 3D CNNs are also able to capture motion details, the addition of optical flow to these networks always benefits classification accuracy. This is likely due to the recurrent refinements these methods use[8]. Carreira and Zisserman (2017) showed that for the I3D model, classification accuracy solely based on motion patterns is comparable to that of only images. As images are a depth of 3, RGB, while optical flow is a flat image with the image values only indicating the rate of change between frames, this indicates that for these classification tasks no fine-grained colour or texture patterns are needed to separate the target classes accurately.

In this section, an overview was provided of some of the difficulties of modelling the spatiotemporal dimension which is frequently obtained by introducing new model architectures using a supervised action classification task to learn a discrim-

[4] Zach et al. (2007)
[5] Wedel et al. (2009)
[6] Pérez et al. (2013)
[7] Fortun et al. (2015)

[8] Karpathy et al. (2014)

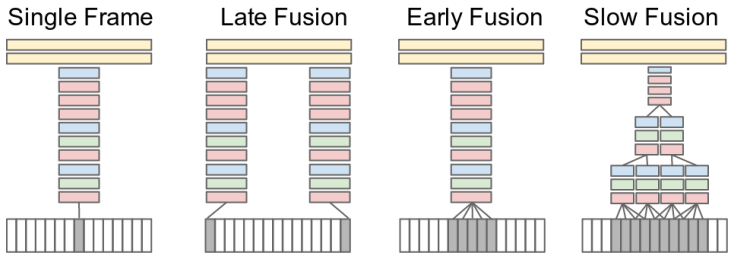


Figure 2.4: A more in-depth illustration of the approaches towards aggregating temporal information as seen in Figure 2.3. *Single Frame* operates on the frame-level and ignores the temporal aspect. *Late Fusion* compares non-consecutive frames and merges the feature-representation right before prediction. *Early Fusion* takes in n-consecutive frames and learns one joint-representation of time and spatial information that incorporates motion. *Slow Fusion* decreases the temporal-depth in stages while merging and comparing different sub-networks. Figure reproduced from Karpathy et al. (2014).

inative feature space. As these provide difficult modelling challenges and require significant computational power, current methods in both action recognition and localisation tasks refrain from training these networks themselves and instead use SOTA feature extractors trained using classification tasks. Subsequently, it is common practice to introduce a new model architecture that uses these extracted features for a particular downstream task. However, arguably this leads to sub-optimal spatiotemporal feature representation for the task of event-localisation due to the different domain these videos originate from compared to action classification tasks and the lesser reliance on spatio-*temporal* features. In the following section we discuss the most popular video datasets used for event classification and localisation tasks.

2.2.2 *Datasets*

To obtain an understanding of which datasets can be used to train a model with the TALL-task in mind, an attempt was made to provide an overview of popular datasets in their difference size, domain and annotation level. Particularly for the field of action localisation and classification tasks. For event-localisation in untrimmed videos, videos need to be temporally annotated which is a time-consuming task and as a result leads to a lower amount of samples per class and the total amount of classes. A simplified overview of the properties these datasets can be categorised by is shown in Figure 2.6. The different aspects of these datasets are now shortly discussed.

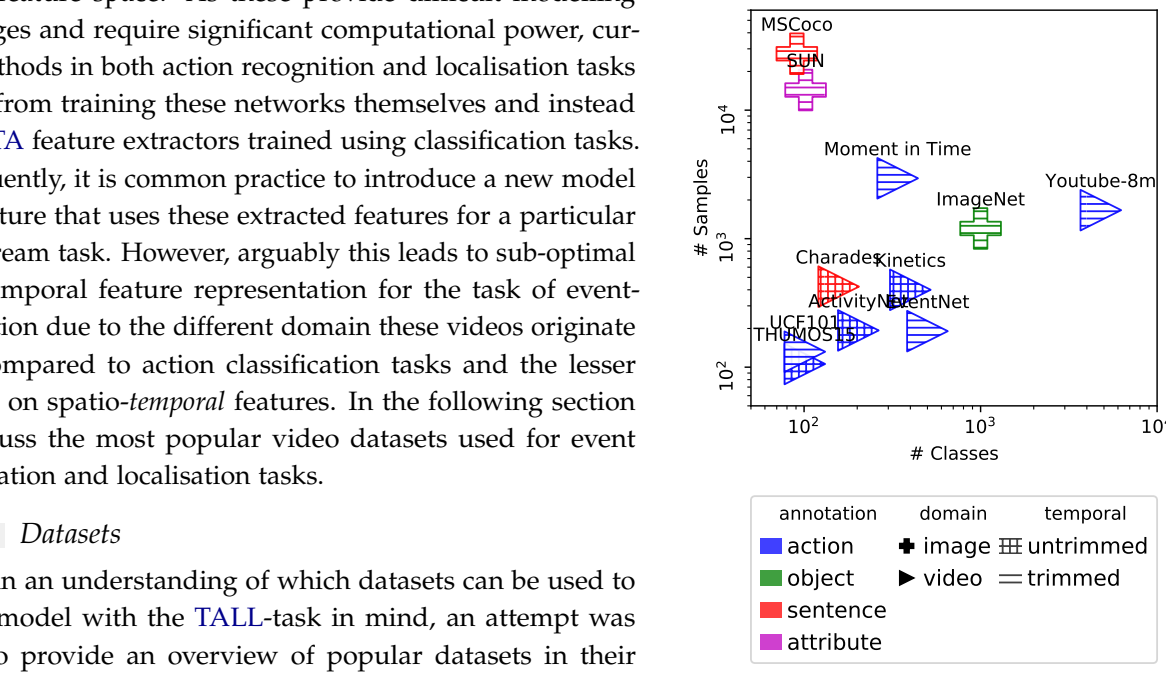


Figure 2.5: An overview of a selection of frequently used datasets in CV with emphasis on event recognition and detection. One can observe significant differences in number of classes and samples per class depending on the annotation-level (*action*, *object*, *sentence*, *attribute*) and domain (*image*, *video*). Videos can either be *trimmed* or *untrimmed*, resulting in a classification or localisation task. More on this shown in Figure 2.6 and 2.9.

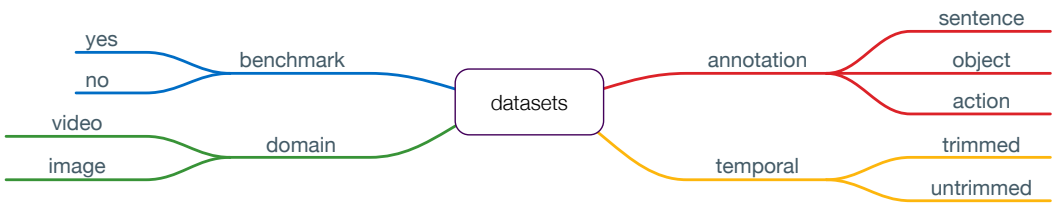


Figure 2.6: Properties of datasets. Colors are corresponding with Figure 2.9. Used for further illustrating how methods rely upon different properties of datasets, including the use of Knowledge Transfer (KT) from the image to video domain.

Domain The datasets used for classification or localisation tasks come from either the image or video domain or a combi-

nation of the two. There is not always a clear correspondence between the task and the domain of the dataset. For example, knowledge transfer from the image to the video domain is common, and some approaches operate on the frame-level in videos rather than n-consecutive frames as the input of the model.

Jain et al. (2015b) are the first to provide an in-depth empirical study of how the recognition of objects within the image domain can be used for the classification and localisation of actions. They show that actions have biases towards specific objects and that the selection thereof is beneficial for action classification tasks. By using 15000 object classifiers trained on ImageNet rather than only a select few image-classes that previous methods frequently used, a significant improvement was obtained on action classification tasks that could also be combined with previous video representation methods. On the frame-level, the likelihood of these object-categories was averaged over the temporal dimension to obtain a complete video representation, which was subsequently combined with additional motion extractors as a representation of the video. An example of how certain events correlate with certain objects can be seen in Figure 2.7.

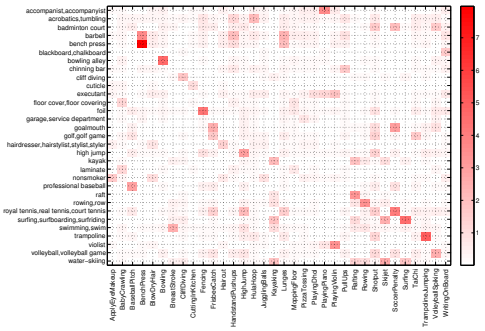


Figure 2.7: An example of how events can be seen as a probability over objects. For example the activity BenchPress frequently contains the object bench-press and barbell. Figure reproduced from Jain et al. (2015b).

Ma et al. (2017) explore whether images from the web can be used as a computationally inexpensive approach towards obtaining training data and improving video classification performance. The benefits of this approach in contrast to using entire videos are the increased variation of; imagery, camera viewpoints, backgrounds, body part visibility and clothing, without the need to deal with redundant or uninformative frames that are apparent in videos[9]. In comparison to videos, images are significantly more subjected to a pre-filtering step in which non-iconic images of a particular action are filtered out such that only the most discriminative part of an action remains. Because of this, presumably Ma et al. find that using unfiltered images from an entire videos are on-par with selecting only a select few images from the image domain to use as training data. Based on this finding, they argue that this can potentially lead to a reduction in annotation labour and can, therefore, more easily scale-up to larger problems.

[9] Ma et al. (2017)

Jain et al. (2015a)[10] attempt to localise and classify actions in an unsupervised fashion by creating an object and action embedding space in which the two are subsequently matched, see Figure 2.8. Jain et al. mention that the limitations of the most common zero-shot approaches in video classification are that the relationships between the unknown and known classes should be predefined by their mutual relationships given by class-to-attribute mappings. These mappings provide underlying lower-level image features that are then shared between unseen and seen classes. They circumvent this limitation by

[10] Jain, M., van Gemert, J. C., Mensink, T., and Snoek, C. G. (2015a). Objects2action: Classifying and localizing actions without any video example. In *Proceedings of the IEEE international conference on computer vision*, pages 4588–4596

extracting features from the softmax layer of a pre-trained ImageNet network as a visual representation while using the textual descriptions of actions and objects from WordNet to perform the matching between the two. This is made possible by the fact that ImageNet uses the concept hierarchy from WordNet, and therefore allows to relate object-classes with their linguistic counterpart (*words*) such that a joint-embedding space can be obtained.

Within the image domain, the most popular dataset is ImageNet[11], a large scale hierarchical database of images that follows the structure of WordNet. To date, it is by far the largest and most diverse image dataset. WordNet[12] is a lexical database of English words in which nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms, therefore named *synsets*. Conceptual-semantic and lexical relations are captured, such that words not close in proximity are semantically disambiguated. ImageNet exploits this fact by structuring the image-classes in a hierarchical tree that consequently allows for additional evaluation methods such as fine-grained vs general classification evaluation or sub-tree removal. Xian et al. (2017) for this reason specifically mention its capability to be used as a dataset for zero-shot evaluation in a broad-setting. As obtaining a broad zero-shot evaluation dataset was one of our objectives in this work, we use this property of ImageNet in Section 5.1.1 to accomplish a similar goal.

[11] Deng et al. (2009)

[12] Miller (1998)

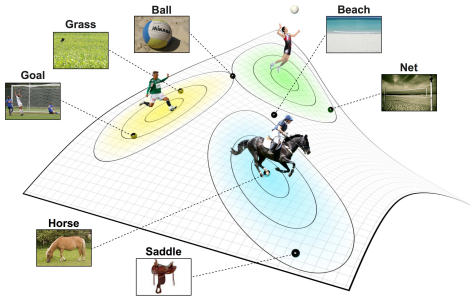


Figure 2.8: Instead of relying upon knowledge transfer by using class-to-attribute mappings, Jain et al. (2015a) embed images and textual descriptions of objects in the same space. Thereafter, they extend their zero-shot approach towards localisation of actions in video with promising results.

Annotation Level & Temporal Videos can be annotated on the *action*- or *sentence*-level, while images are frequently annotated on the *object*- or *sentence*-level. In literature, the notions of *actions* and *events* are frequently used interchangeably. Here, actions are both referring to a narrow domain of sport-actions, as well as to the broader sense of actions, such as ripping paper or shovelling snow[13]. Depending on the annotation-level, the datasets are either used for a localisation or classification task. In Figure 2.9 the relation between the annotation level and whether the videos are trimmed or untrimmed is shown. The action localisation task can be viewed as two consecutive subtasks; the detection of an action and after that the classification thereof. The TALL-task is different from the usual localisation task in that the target classes in action localisation setting are usually fixed and span only up to a few hundred target classes, while in the former it can be described using any natural language text. This formally means that in the TALL-task the matching between videos and text happens in semantic space (the labels consist of n-dimensional *continuous* embeddings), whereas in general localisation tasks this matching is accomplished in label-space (labels only indicate whether a particular class is apparent in the video with a one or multi-hot *discrete*

[13] Kay et al. (2017)

encoding).

Popular sentence annotated datasets within the image domain are MSCoco[14], Flickr30k[15], OpenImages[16]. Examples of datasets within the video domain that are annotated on the sentence level are; Charades-STA[17], TACoS[18], and on the action level are UCF101[19], Sports-1M[20], HMDB51[21], AVA[22], Kinetics[23]. However, whereas within the image domain large datasets can have millions of training examples, temporally annotated datasets consider themselves large with only a few hundreds or thousand videos[24] (Figure 2.5, Table 2.1).

Large temporally annotated datasets naturally exist within the domain of videos, such as movies with subtitles. However, a significant difference is that these subtitles describe not what visually happens within the video, but rather represent what is being said by the actors that is to a large extent independent thereof. Therefore, with the introduction of the TALL-task, the Gao et al. specifically added temporal annotations to the already existing Charades dataset in order to obtain a larger temporally annotated dataset than was currently available which they called Charades-STA.

[14] Lin et al. (2014)
[15] Plummer et al. (2015)
[16] Krasin et al. (2017)
[17] Gao et al. (2017)
[18] Regneri et al. (2013)
[19] Soomro et al. (2012)
[20] Abu-El-Haija et al. (2016)
[21] Kuehne et al. (2013)
[22] Gu et al. (2017)
[23] Kay et al. (2017)
[24] Regneri et al. (2013), Gao et al. (2017)

dataset	Citation	Release	Source	Focus	Annotation	Avg. Duration	# Videos	# Categories	Untrimmed?	Benchmark?
THUMOS14	Jiang et al. (2014)	'14	YouTube	Human Actions	Single-label, Temporal	4.6s	18k	101	☒	☒
THUMOS15	Gorban et al. (2015)	'15	YouTube	Human Actions	Single-label, Temporal	4.6s	23.7k	101	☒	☒
Charades	Sigurdsson et al. (2016)	'16	Home-Recordings	Human Daily Activities	Temporal	30s	10k	157	☐	☐
Youtube-8M	Abu-El-Haija et al. (2016)	'16	YouTube	General	Multi-label	120-500s	$8.2e^6$	4716	☐	☐
ActivityNet1.3	Caba Heilbron et al. (2015)	'16	Web-search	Human Actions	Multi-label	5m-10m	28k	200	☒	☒
Kinetics	Carreira and Zisserman (2017)	'17	YouTube	Human Actions	Single-label, Temporal	10s	306k	400	☒	☐
AVA	Gu et al. (2017)	'17	Movies	General Daily Life	Multi-label, Temporal	15m	192*	80	☒	☐
Moments in Time	Monfort et al. (2018)	'18	Web-search	General	Single-label	3s	$1e^6$	339	☐	☐

Table 2.1: Summary of major action recognition datasets.

Besides supervised approaches for learning visual representations with labels used as the target, also unsupervised approaches exist in the literature that do not rely on the aforementioned datasets. Wang and Gupta (2015) use an unsupervised approach for learning video representations by designing a Siamese-triplet loss network with a ranking loss-function to train a visual representation. Whereas in previous work either an auto-encoder was used to reconstruct static images[25], they build their method upon the realisation that humans instead learn their visual representation of the outside world through dynamic sensory input (the slight variation of visual input over time). In the triplet loss-function, they enforce that two frames close to each other should have a similar representation while a randomly sampled third frame should be dissimilar. On the PASCAL VOC[26] dataset, they obtain almost similar results in this unsupervised version with an accuracy of 52.0% compared to 54.4% for the supervised alternative.

Huang et al. (2016) attempt to learn attributes from images by using semantic clustering. Their realisation is that large amounts of human labelling is expensive and that attributes in the label space are not necessarily discriminative in the feature space. Instead, they predict attributes that are representative

[25] Wang, X. and Gupta, A. (2015). Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802

[26] Everingham et al. (2010)

and discriminative by introducing artificial clusters while maximising their separability in feature space. For more details, we defer to Huang et al. (2016), but the main point is that to learn a discriminative feature space for videos or images, unsupervised approaches have been proposed with some indications of initial success.

BENCHMARK Popular datasets for evaluating video classification in untrimmed videos are THUMOS14 and THUMOS15, Sports-1m and ActivityNet. Trimmed video datasets include; UCF101, HMBD51 and more recently also ActivityNet. Although these benchmark datasets have separate training-sets, to a large extent current models use KT techniques from models trained on significantly larger datasets to obtain SOTA performance (e.g. [27,28]). The reliance upon KT techniques partially defeats the purpose of these evaluation methods as this shifts the focus to properties that are independent of the model. With larger datasets being released every year, this specifically affects the ability to compare less recent work with the currently released work as Deep Learning (DL) approaches frequently benefit significantly from larger datasets.

[27] Carreira and Zisserman (2017)
[28] Xie et al. (2017)

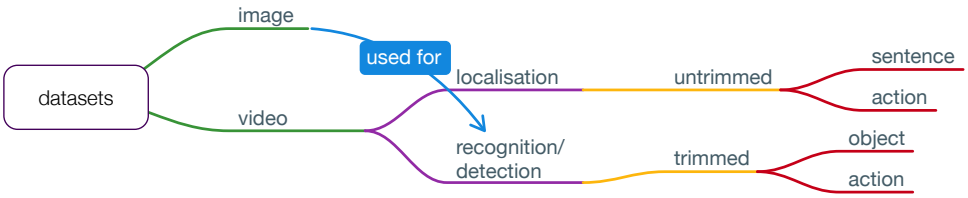


Figure 2.9: General trends towards which datasets are used for either the localisation or recognition/detection tasks.

Idrees et al. (2017) describe how the popular THUMOS challenge and datasets were created with the purpose of improving video understanding applications. In the recent THUMOS challenge of '14 and '15, the focus was shifted towards untrimmed rather than trimmed videos. Idrees et al. argue that for a less-artificial evaluation setup of action localisation and classification also background videos should be included in which a similar scene is apparent of a particular action without the action actually taking place. This way no longer simplify the presence of certain objects within the frame can be used to correctly predict the temporal window of an event or classification thereof. For example, a piano standing in the background does not imply it is being played. Therefore the THUMOS '14 and '15 both started to include background videos.

Idrees et al. discuss future improvements for obtaining datasets for improving video understanding and recommend to place emphasis on actions and activities performed by humans individually or in groups combined with dense annotations

for; objects, actions, scenes, attributes and inter-relationships between objects, actions and the environment. To add the necessary annotations, they recommend using WordNet which allows the modelling of structured knowledge to relate the different nouns, verbs and adjectives needed to understand language. For this Idrees et al. note that it is important to consider the trade-off between *consistency* and *diversity*. Consistency requires that the labels (e.g. objects or actions) are reused across videos which helps to relate videos that share the same labels, whereas diversity is meant to increase the use of the vocabulary such that also for example synonyms are used (e.g. man, person). Idrees et al. describe that WordNet can be used to relate hyponyms and hypernyms to each other to transfer attributes and properties in order to save time and effort while generating richer and dense annotations for a large video dataset. We will see in Section 5.3.4 and 5.3.5 that the TACoS and the Charades-STA datasets that were used for evaluating the TALL-task did not contain a large vocabulary. Therefore, this evaluation setup favors consistency to a large extent over diversity. However, arguably diversity is important for the evaluation-setup of the task of event-localisation given natural language due to the many complex relationships between words including synonyms, antonyms and hyponyms. This is further discussed in the Discussion Section (7).

Instead of improving the video annotation method to improve video understanding, in our approach we attempt to improve the representation of language directly such that the relationships between e.g. the hyponyms and hypernyms between words already contain this similarity. This could potentially uplift the dependency upon costly annotated datasets.

Recently, critique has been brought up regarding the current methods and reliance's upon specific datasets in action classification and localisation tasks. Including (1) the dependency upon trimmed videos for video classification which is in contrast to the general nature videos occur in[29] and (2) although action localisation tasks and datasets partially resolve this problem these datasets are still confined to a relatively narrow domain with only a small number of classes[30]. This poses questions about the generalisability of these methods concerning the goal of achieving *general* video understanding. An example of a practical application that would require such a feature could for example be video-playback of a specific event through voice commands for streaming services.

[29] Idrees et al. (2017)

[30] Gao et al. (2017)

2.2.3 *Computational Efficiency*

When working in the domain of vision the computation efficiency of the proposed model architecture is important for many downstream applications. There have been many works proposed that specifically focus on this aspect for the localisa-

tion of events in videos. What makes this especially difficult is the added temporal dimension and large variety of duration of events that require a more expensive search operation. A common but slow approach towards this problem is to first have an action proposal method suggest where a possible action occurs which happens using multiple temporal resolutions using a sliding window approach while utilising extracted features from CNNs trained on a different task. This reduces the amount of training time needed immensely as training visual representation from scratch is one of the most computationally expensive operations in CV. After the action proposal network there follows a proposal classification step. To date this is still the status quo[31], however many computational improvements have been proposed. As computational efficiency is such an essential part of any video search or localisation application, these are now shortly discussed.

[31] Xu, H., Das, A., and Saenko, K. (2017). R-c3d: Region convolutional 3d network for temporal activity detection. *arXiv preprint arXiv:1703.07814*

Xu et al. (2017) improve the standard action proposal and classification pipeline by sharing the parameters between both individual segment proposal and classification networks which decreases the computational cost and allows for end-to-end learning. Xu et al. argue that the use of feature extraction methods trained on a different task could lead to sub-optimal representations for the localisation of activities in diverse video domains, resulting in inferior performance. In addition, during the action proposal stage, the videos need to be scanned at a variety of temporal scales using sliding window approaches, leading to poor computational efficiency. In a similar fashion as the Faster R-CNN object detection approach, they compute 3D CNN features and propose temporal regions likely to contain activities, after that these feature regions are pooled and used to predict activity classes.

Caba Heilbron et al. (2016) mention that applying action classifiers at every time location and multiple temporal scales is unfeasible in a large-scale video analysis application which in part is causing that activity detection in large-scale video collections remains relatively unexplored. Caba Heilbron et al. attempt to improve the initial spatiotemporal proposal step. The benefit of spatiotemporal proposals that still include time and space is that this allows separating co-occurring and overlapping events. Criteria for success for this initial step are (a) high recall and relatively good precision but more importantly that (b) it should produce these proposals quickly. However, Caba Heilbron et al. show that when proposals are made only in time excluding the space dimension, limited performance losses are obtained while it improves the scaling of this method to significantly larger scale video datasets such as THUMOS and ActivityNet.

In Figure 2.3, an overview was provided of the different ways optical flow is included in popular video representation

learning architectures. Initially, calculating the optical flow of video segments prevented any of the methods that relied upon this to run in real-time as these methods could not be applied in real-time. Zhang et al. (2016) tried to improve upon this by replacing the optical flow- with motion-estimation. Their key realisation is that motion and optical flow estimations are inherently correlated. Motion vectors are being used in video compression by exploiting how one macroblock of an image moves across time to reduce the bit rate in video compression. Therefore, in contrast to optical flow estimation, the movement patterns are coarse and not precise. By using the encoded video directly, and learning how this corresponds with the optical flow through a parameterised model, significant improvements were obtained including real-time motion estimation that was much closer to the precision found in optical flow.

Nguyen et al. (2017) propose a weakly supervised deep neural network that selects a sparse subset of frames for the task of action recognition. Nguyen et al. introduce a loss-function that penalises both the classification error while also encouraging sparse frame selection to predict the appropriate class. As a result, instead of analysing the whole video before making the class prediction, only a subset of the frames in the video are observed before making the prediction. As argued, this is especially important in untrimmed videos in which only a small fraction of the frames actually contain an action of interest. SOTA was obtained on the THUMOS14 and ActivityNet1.3 dataset indicating the success of this approach.

2.3 *Language Representation - Modeling Difficulties*

Thus far we have discussed some of the identified problems with learning an appropriate *video* representation, without mentioning the efforts done to improve the representation of *language*. However, when it comes to searching or localising specific events denoted in natural language text, this is arguably equally important. Although the TALL-task is new and the inclusion of natural language has been relatively unexplored[32], there are a few works in literature that take a language first-approach when it comes to video or image understanding which are discussed now.

[32] Gao et al. (2017)

De Boer et al. (2017) attempt to provide an overview of the current challenges of zero example video event retrieval. In their work they focus on the TRECVID MED[33] challenge; retrieving relevant video clips given only a precise textual description of a complex event. As such it shares a similar objective as is attempted here. They emphasise on two challenges of designing an effective system for zero example complex event retrieval in video; the *vocabulary*- and *concept* selection-

[33] Over et al. (2015)

challenge.

For the concept selection challenge, the objective is to find the right concepts to pre-train on. Not all concepts are deemed equally useful for the transfer of knowledge to unseen classes, for example, ImageNet contains a large variety of dog breeds unimportant for most tasks. De Boer et al. further expand on existing literature by showing that including high-level concepts (e.g. events) are generally more important than low-level concepts (objects) for accurate performance for the task of event recognition given any textual description. Improvements are however obtained when combining the two. Earlier work already demonstrated that task-specific concept-selection in contrast to selecting all concepts lead to improved performance[34]. Besides, it can be expected that whereas some concepts have clear visual correspondences others may not, which potentially limits their usefulness to be matched with visual-features.

[34] Mazloom et al. (2013)

The second challenge is the concept selection challenge in which words in the query have to be matched with the pre-trained concepts obtained in the vocabulary challenge. Liu et al. (2007) provide five different ways to achieve this, including; using a KB, selecting the most relevant concepts in the training-set based on machine learning methods and incorporating relevance feedback from the user click behaviour to re-order the search results over time. Our approach here can be considered an instance of an approach towards solving the vocabulary challenge that uses a KB. By creating systematic language embeddings with the usage of a KB, relating the seen vocabulary during training time to the unseen vocabulary during testing is expected to improve.

De Boer et al. designed a system that augments and or changes the User User Query (UQ) into an underlying System Query (SQ). For each word in the UQ that is not in the training-set the closest word in the word2vec vocabulary are averaged till the average cosine similarity between the UQ and actual word-embedding does not increase any more. Only words are included that have a cosine similarity above a certain threshold. This way the problem of *query drift* is partially prevented, in which replacing words used in the UQ with multiple words in the SQ degrades the actual similarity between the UQ and SQ. In our work, instead of doing query expansion from UQ to SQ and keeping the representation of language fixed, an attempt is made to encode the essential relationships between *close* concepts directly in the language representation.

When dealing with the vocabulary challenge, De Boer et al. see that concepts can relate to each other in two important ways. On the levels of *granularity* and *complexity*[35]. The linguistic terms for the former are hyponyms (spoon is the hyponym of cutlery) and hypernyms (colour is the hypernym of red). At the same time, events can be described on different levels of

[35] De Boer et al. (2017)

complexity, on the object level, basic actions, activities, interactions or on an even higher level of complex activities involving people interacting with other people and objects. As these relationships are already an essential part within KB, using a KB directly to solve the vocabulary challenge could make sense. The concept selection challenge, however, remains an issue using this approach as the visual cues that are matched with language in the cross-modal embedding space is ultimately dependent upon the selection of visual examples.

Whereas significant progress has been made in learning visual-semantic embeddings for zero-shot learning, relating images to text is still far from a resolved challenge[36]. Reed et al. (2016) take a language first approach in an attempt to solve the particular challenge of matching images with fine-grained textual descriptions of image-classes even for zero-shot test cases. Although Reed et al. focus on zero-shot *image* recognition and retrieval in a *narrow* domain of birds, the primary objective is shared in which language rather than vision is the focus towards improving zero-shot event recognition. The datasets that are available for training such a model are, however, limited. Wikipedia is one of the largest sources of textual data but is not visually focused, whereas selecting only the visually centred subset thereof once again makes the dataset too small. Reed et al. (2016) hypothesise that using larger-capacity text models are required for high zero-shot text embedding methods, which increase the reliance on larger visually centred datasets: images with multiple visual descriptions. Their approach leads to SOTA performance on the CaltechUCSD Birds 200-2011 dataset.

[36] Reed et al. (2016)

Another difficulty Reed et al. (2016) face is how to train models end-to-end in which both the visual and textual representation are being learned jointly. The CUB and Flowers datasets were used with Amazon Mechanical Turk (AMT) being used to obtain ten visual descriptions for each image. By using a CNN-RNN model architecture for learning the representation of language on the character level, robustness for typos was obtained. The learning objective was formulated as a surrogate objective function where the textual and visual representation of training examples from the same class were forced to be the same. SOTA performance was obtained using this approach which indicates that current class-to-attribute mappings are not required to obtain high performance given this particular task[37]. However, the careful selection of visually grounded image descriptions and narrowly confined domain the images are situated in within the CUB and Flowers dataset make it difficult to generalise these findings to the domain of event-localisation given natural language text as this would rely more on coarse rather than narrow video understanding.

[37] Reed et al. (2016)

In 2017 ConceptNet 5.5 was introduced with the intention

to further improve language machine learning applications such as methods towards obtaining word embeddings. The multilingual KB was explicitly designed to represent general knowledge involved in understanding language by incorporating knowledge from multiple KBs while also adding additional expert knowledge. Speer et al. (2017) note that when ConceptNet is combined with word embeddings acquired using distributional word semantics it provides applications with additional understanding that can not be obtained using solely distributional embeddings, which is in line with our objective in this work. The dataset consists of ternary relations $< subject, relation, object >$, and includes inter-language relations. In their follow-up paper[38], the authors go into detail how to obtain improved language embeddings using ConceptNet in a method called retrofitting. This resulted in the already mentioned Numberbatch in the Introduction, the current SOTA in language embeddings. Our approach resembles the inclusion of both the knowledge contained in ConceptNet and distributional word embedding approaches. The differences between their approach and ours is further discussed in Section 2.5.

[38] Speer and Lowry-Duda (2017)

2.4 *Remarks & Sub-Conclusions*

In the previous sections we have explored some difficulties in literature in obtaining an effective video representation for video understanding tasks. In specific, the lack of task-specific end-to-end trained video representations and large-scale video datasets combined with efficient models for event-localisation are frequently brought up as the most difficult challenges currently limiting the speed in which the field can progress. More importantly, the representation of video is still being learned using only a select few target event classes represented by a one hot encoding which greatly simplifies the usage of language. Arguably for the task of event-localisation a thorough understanding of how visual cues correspond with language is needed. Gao et al. (2017) in their approach represent language using general language embeddings obtained using distributional word vector methods. In our work, we attempt to improve the representation of language specifically with this task in mind. The next section provides the reader with a general introduction to GCs and how they can be potentially used to obtain language embeddings specifically tailored to the TALL-task.

2.5 *Graph Convolutions & Language Embeddings*

GCNs have recently become popular with two main directions to their applicability; link prediction and node classification[39,40]. Graph Convolutional Networks models share the

[39] Goyal and Ferrara (2017)

[40] Cai et al. (2017)

objective of taking in a graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and produce node-level output representations $\mathcal{Z}$ in an attempt to minimise the reconstruction error of the graph. Therefore, $\mathcal{Z}$ can be considered a condensed node-embedding feature representation of the graph. These models share a feature description x_i for every node i represented in the matrix $\mathcal{X}$ of size $\mathcal{N}x\mathcal{D}$ with an additional adjacency matrix $\mathcal{A}$ that for each node contains a multi-hot encoding representing the relationships to neighbouring nodes.

For our task, the objective is to find suitable node-embedding representations of the nodes in the ConceptNet KB in which each node roughly corresponds with one word in the English language. As a result, node classification methods could at least in theory be applicable here as these methods have been shown to efficiently encode complex graphs to low-dimensional word vectors. In unsupervised classification approaches, the heuristic can be used that the local neighbourhood of a particular node should have a more similar feature-representation than nodes further apart in the graph. When it comes to obtaining low-dimensional language embeddings, the current SOTA according to many of the intrinsic bench-marking metrics (see 6.2) is actually based on ConceptNet[41,42] which is a knowledge graph that connects words and phrases with labelled edges. Numberbatch uses the technique called *retrofitting* in combination with additional efforts towards decreasing the number of biases, including gender bias and ethnic biases. A comparison to other language embeddings on a select few intrinsic evaluation methods and bias-metrics can be seen in Figure 1.2.

[41] Havasi et al. (2007)

[42] Speer and Havasi (2013)

Now follows a short comparison between our proposed solution and the retrofitting method.

2.5.1 *Retrofitting vs Graph Convolutions*

Retrofitting In retrofitting lexical relational sources are used to obtain higher quality semantic word vectors as a post-processing step through belief propagation which can be applied to any vector training model[43]. Therefore this approach can also be applied on any pre-trained word-embedding. The prerequisite is a set of words and their representations given by a vocabulary $V = w_1, \cdots, w_n$ and an ontology Ω that encodes the relationships between the words which is represented as an undirected graph $\mathcal{G}$ consisting of vertices and edge pairs (V,E) with one vertex for each word type and edges $(w_i, w_j) \in E \subseteq VxV$. The matrix $\vec{Q}$ is a collection of vector representations $\vec{q_i} \in \mathcal{R}^d$ for each $w_i \in V$ where d is the length of the word vectors. Our objective is to learn the matrix $Q = (q_1, \cdots, q_n)$ such that the columns are close to their counterparts in $\vec{Q}$ in addition to the adjacent vertices in Ω. The distance between a pair of vectors is defined by the Euclidean

[43] Faruqui et al. (2014)

distance, such that the minimisation objective becomes,

$$\Psi(Q) = \sum_{i=1}^{n} \left[\alpha_i \|q_i - \hat{q}_i\|^2 + \sum_{(i,j)\in E} \beta_{ij} \|q_i - q_j\|^2 \right] \tag{2.1}$$

with α and β determining the weight between the two loss-function components. As Ψ is convex in Q, this can be solved using a system of linear equations. Faruqui et al. (2014) state that running the following update rule, which is the first derivative of Ψ with respect to q_i, and equating it to zero for ten iterations converges within 10^-2 distance from the optimal solution within 5 seconds given the dataset sizes displayed in Figure 2.10.

$$q_i = \frac{\sum_{j:(i,j)\in E} \beta_{ij} q_j + \alpha_i \hat{q}_i}{\sum_{j:(ij)\in E} \beta_{ij} + \alpha_i} \tag{2.2}$$

The first term in Equation 2.1 weighted by α can be interpreted as a prior on the representation of the word-embeddings.

Lexicon	Words	Edges
PPDB	102,902	374,555
WordNet_{syn}	148,730	304,856
WordNet_{all}	148,730	934,705
FrameNet	10,822	417,456

Figure 2.10: Approximate size of the graphs used to apply retrofitting on. Table reproduced by Faruqui et al. (2014)

Graph Convolutions The specific GCN we are using is called GraphSAGE. A general introduction specifically designed to compare this method with the retrofitting technique is given now, for a more thorough overview of the GraphSAGE we defer to Section 3.3.

In contrast to the standard GCN that learns the node-level feature representation in a transductive setting in which during training and testing the same graph is being used, GraphSAGE sets itself apart by learning node feature representations in an inductive setting. Compared to the transductive setting, in the inductive framework one must learn to recognise structural properties of a node's neighbourhood that reveals both the node's local role in the graph as well as its global position[44]. We expect that the added information about the node's global role in the graph that is not included in the retrofitting setup further improves the zero-shot learning performance due to its added structure.

[44] Hamilton et al. (2017)

Whereas in the transductive setting information is propagated from the local neighbourhood only, the added requirement to incorporate global information of the entire-graph increases the computational cost significantly. For computational efficiency Hamilton et al. propose the SAmple and aggreGatE method (GraphSAGE). Here local neighbourhood information is first sampled by a neighbourhood function $\mathcal{N}(v)$ that takes in a vertex and returns a random one from the neighbourhood which is defined as being within n hops distance. One hops is being equal to one edge-traversal starting from a particular node of interest. As the retrofitting technique does not feature any stochastic process, this marks one difference between both methods. Subsequently, depending on the number of hidden

layers, information is aggregated from the local neighbourhood using a variety of parameterised aggregator functions. The term *aggregator function* is being used to describe any function that combines the existing node feature representation h and incorporates additional local neighbourhood information within this representation. Starting with the node feature embeddings x (the language embeddings which are equivalent to $\hat{q}_i$ in the retrofitting example) which is initially set to be equal to h, the objective is to learn a feature representation $\mathbf{z}_u$ through a chosen aggregator function to minimise the following objective function:

$$J_{\mathcal{G}}(\mathbf{z}_u) = -\log(\sigma(\mathbf{z}_u^T \mathbf{z}_v)) - \mathcal{Q} \cdot \mathop{\mathbb{E}}_{v_n \sim P_n(v)} \log(\sigma(-\mathbf{z}_u^T \mathbf{z}_{v_n})) \quad (2.3)$$

Where $\mathbf{z}_u$ is the result of one or multiple layers of parameterised node feature aggregation functions with a fully connected layer followed by a non-linearity. The global position of the node is added indirectly to the node-feature representation by enforcing that negative samples ($\mathbf{z}_{v_n}$ sampled by $\mathcal{P}_n(v)$) should be dissimilar. It is important to note that $J_{\mathcal{G}}$ updates and uses the feature representation of $\mathbf{z}_u$ instead of the original node feature representation as was the case in the retrofitting example ($\hat{q}_i$). As such, the representation of $\mathbf{z}_u$ can become significantly different from the original feature representation ($\hat{q}_i$ in the retrofitting example, x in the GraphSAGE example). Hence, the main difference between the methods of retrofitting and GraphSAGE is that information in the latter is aggregated using (1) a parameterised function and non-linearity optimized by gradient descent using (2) a sampling technique instead of the entire neighbourhood and most importantly (3) is dependent upon the learned feature representations instead of the initially learned feature representation from the distributional word-embedding method.

3 Related Work

In our work we use the TALL network architecture introduced by Gao et al. (2017) in order to temporally localise events given extracted video features and language embeddings obtained using the GraphSAGE model architecture introduced by Hamilton et al. (2017). Therefore an overview of these works is now discussed in depth, in addition to some work centred around the evaluation of word-embeddings and zero-shot evaluation methods that are used in the methods and experimental setup section.

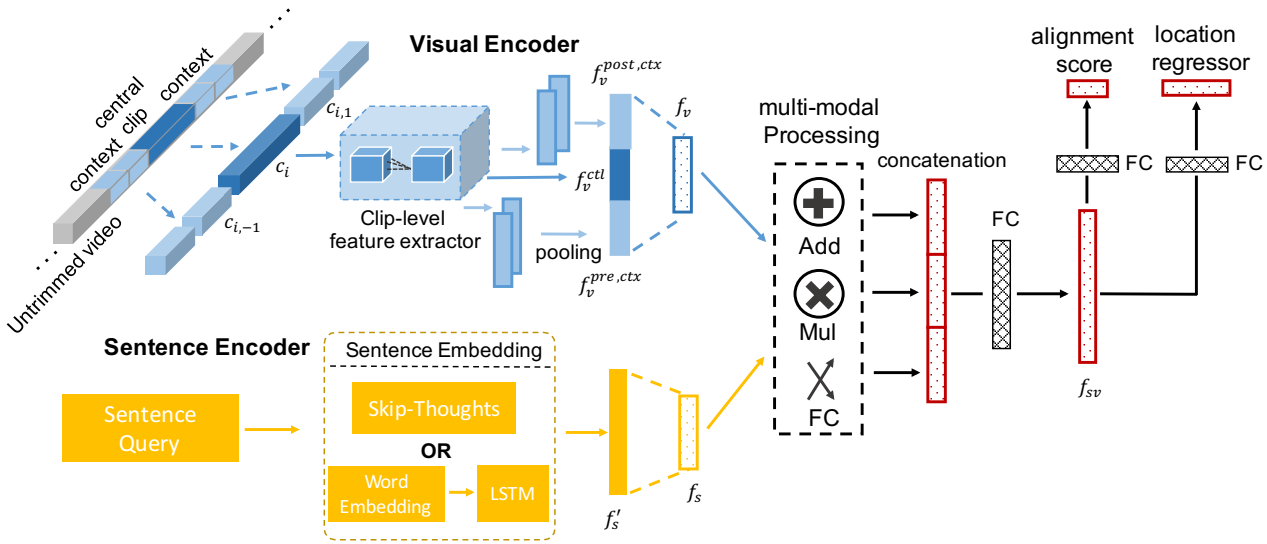


Figure 3.1: The TALL module architecture; Cross-modal Temporal Regression Localizer (CTRL) architecture. CTRL contains four modules: a visual encoder to extract features for video clips, a sentence encoder to extract textual embeddings, a multi-modal processing network to generate combined representations for the visual and text domain, and a temporal regression network to produce alignment scores and location offsets. Figure and its description reproduced from Gao et al. (2017).

3.1 TALL Model Architecture

Gao et al. (2017) mainly focus in their work on finding a suitable solution for their earlier introduced TALL-task by obtaining an appropriate model to project language and visual features into a common cross-modal embedding space for accurate temporal localisation. As this is a newly introduced task, no earlier benchmark scores exist[1]. They address the problems on how to localise events of significantly different lengths and how to use natural language text in the process as opposed to using only a small list of pre-defined event classes in a completely supervised fashion. The former can be characterised as a computational problem as densely sampling features and aggregating information on multiple time-scales increases the search space significantly. The latter is changing the modelling formulation from a completely supervised and balanced classification problem into one with a more complex cross-modal embedding space in which both the labels (text)

[1] Gao et al. (2017)

and video are projected into a continuous space in which the matching happens between the two.

An overview of the model architecture that was used to address these issues can be seen in Figure 3.1 and is introduced as the Cross-modal Temporal Regression Localizer (CTRL) model architecture. The most important components are the; *visual encoder*, which maps video segments into visual features f_v, *sentence encoder*; that maps text to textual features f_s, a *multi-modal processing unit*; that combines the two modalities into one joint representation f_{sv}, and lastly the *temporal localisation regression network*; that consists of two separate loss-functions that optimize for the similarity between the two modalities (*alignment score*) and the predictions of the start and end time of the particular event given the visual input (*location regressor*).

3.1.1 *Modality Feature Extraction*

Visual features are extracted from untrimmed video clips by cutting the video using a multi-scale sliding window with 80% overlap with [64, 128, 256, 512] frames using the C3D modal architecture trained on the Sports1M dataset. As the C3D clip-level feature extractor model takes in only 64 consecutive frames, frames are sampled uniformly when the sliding window length exceeds this amount. Context clips besides the central clip are given to the model with the intention of improving the ability to localise the temporal boundaries using the location regression loss (Section 3.1.3). As events can be of arbitrary length, multiple context clips are average pooled to accommodate longer video clips ($f_v^{pre,ctx}$ and $f_v^{post,ctx}$). Language embeddings are obtained either on the word- (Word2Vec) or sentence-level (Skip-Thoughts) which are both obtained using networks that rely upon the distributional hypothesis. The authors accredit the lower score they obtained for training sentence embeddings by using a LSTM aggregator function on the word-level due to the limited dataset size (Figure 3.2).

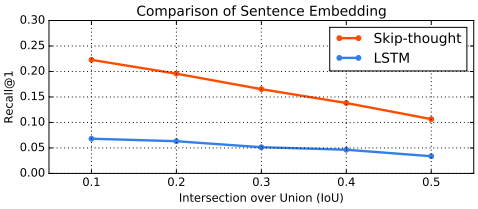


Figure 3.2: Significant performance differences were observed using either a learn-able sentence encoder from the word-level or pre-trained sentence-embeddings. Gao et al. (2017) accredit the observed difference due to the limited dataset size that make the training on the word-level unfeasible. Figure reproduced from Gao et al. (2017).

3.1.2 *Sampling Training Examples*

As the multi-scale sliding window approach towards obtaining visual features does not lead to video-clips that overlap entirely with the temporal sentence annotations windows, Gao et al. (2017) mine training examples only if (1) the IoU between the extracted video-segment and temporal sentence annotation window is greater than 0.5, the non-nIoL is smaller than 0.2 and (3) only one sliding window clip can be assigned to one sentence description. In Figure 3.3 an illustrative example is shown of the different terms. While using IoU is standard practice when dealing with a localisation task, the addition of nIoL was used to ensure minimal overlap between the visual features and other sentences for the *location regression* loss. The latter is a significant problem with the used TACoS and

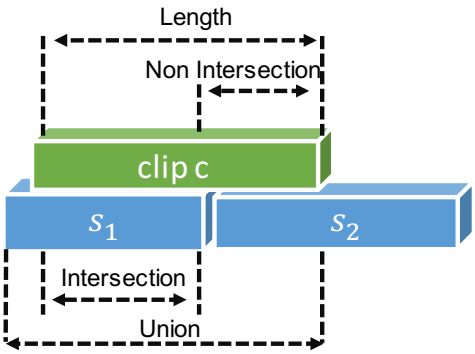


Figure 3.3: Illustrative schematic overview of the different terms used to extract positive training examples and Intersection over Union (IoU)/non-Intersection of Length (nIoL) calculations. Figure reproduced from Gao et al. (2017)

Charades-STA datasets as the temporal sentence annotations are dense and often directly follow each other.

3.1.3 *Loss Functions*

The multi-task loss-function consists of two loss-functions and one hyper-parameter controlling the trade-off between the two. The visual-semantic alignment score loss L_{aln}, the clip location regression loss L_{reg} and a weighting factor α:

$$L = L_{aln} + \alpha L_{reg} \tag{3.1}$$

The alignment loss encourages aligned clip-sentence pairs to have positive scores and misaligned pairs to have negative scores,

$$\begin{aligned} L_{aln} = \frac{1}{N} \sum_{i=0}^{N} [\alpha_c \log(1 + \exp(-cs_{i,i})) + \\ \sum_{j=0, j \neq i}^{N} \alpha_w \log(1 + \exp(cs_{i,j}))] \end{aligned} \tag{3.2}$$

where N is the batch size, $cs_{i,j}$ is the alignment scores between the sentence and video clip, α_c and α_w denote the weight of positive (aligned) and negative (misaligned) examples respectively. The regression loss L_{reg} is only calculated for the aligned clip-sentence pairs,

$$L_{reg} = \frac{1}{N} \sum_{i=0}^{N} [R(t^*_{x,i} - t_{x,i}) + R(t^*_{y,i} - t_{y,i})] \tag{3.3}$$

where sentence s_j contains the temporal start and end time, τ^s_j and τ^e_j. The $R(\cdot)$ function is a smooth L_1 function. For the accurate temporal localisation, two configurations were tested, a parameterised and non-parameterised version to predict the centre and length of the action. Both cases are symbolised by the x and y in the L_{reg} loss-function, which are represented by p and l for the parameterised version,

$$t_p = (p - p_c)/l_c, t_l = \log(l/l_c) \tag{3.4}$$

where p and l are the clip's centre coordinate and clip length. For the non-parameterised version x and y are represented by s and e.

$$t_s = s - s_c, t_e = e - e_c \tag{3.5}$$

3.1.4 *Evaluation Setup*

The performance of the described model-architecture was measured on the TACoS and Charades-STA datasets in terms of IoU $\in \{0.5, 0.3, 0.1\}$ at recall@$\{1, 5\}$ and compared to a random baseline. The score was calculated as a percentage of the overall sentences of which the IoU is larger than the given

$\{0.5, 0.3, 0.1\}$ ranked by IoU at the given recall cutoff $\{1, 5\}$. The random baseline was created by randomly selecting n windows from the test sliding windows of which the IoU at the different recall cutoffs was then calculated. Therefore the scoring function can be formalised as,

$$R(n, m) = \frac{1}{N} \sum_{i=1}^{N} r(n, m, s_i) \tag{3.6}$$

where N is the total number of queries and $R(\cdot)$ is the average performance among all sentences where the $R@n$ (sorted in descending order is higher) than the selected IoU@m given a sentence s_i.

3.1.5 *Observed Difficulties*

Gao et al. (2017) mention that long query sentences increase the chance of failure presumably due the sentence embeddings not being discriminative enough. Significant performance losses were observed going from word-level to sentence-level embeddings through a parameterised model (Long Short Term Memory (LSTM)) when compared to using pre-trained sentence-level embeddings on a different task (Figure 3.2). The reason Gao et al. (2017) give for this is the limited dataset size of only 127 videos. Therefore this limitation is inherently a dataset-size problem rather than a modelling limitation. In the Background Section (2.2.3) the problem of query drift was discussed in which longer queries result in less discriminative sentence representations. Although Gao et al. do not rely on matching the UQ with a SQ as was the case in the work of De Boer et al. (2017), still the same problem of query drift is likely to occur as longer queries result in less discriminative feature-representations due to the averaging over multiple words.

The second limitation that was mentioned is that when the same motion was used but with different objects (e.g. putting a cucumber *or* knife on a cutting board) the model had difficulty in distinguishing the two. This indicates that additional focus should be dedicated to the accurate localisation of specific objects in the visual representation. As a pre-trained CNN model was used trained on the Sports1M dataset, this problem can partially be accredited to the domain-shift going from the domain of sports to activities in homes which requires a different feature-representation to separate the events effectively.

3.2 *The Addition of Language in Action Localisation*

Thus far in most approaches, visual input is being matched with its language counterpart consisting of only a predefined and small list of event-classes described in a single word. In the models that are being used in these approaches, the visual

input is used as input of the network with the target language representation being represented using a one-hot-encoded representation of the corresponding language "classes", therefore greatly simplifying the language. To go from a well-constrained representation of language to one that closely resembles natural language text, a few approaches frequently used in the literature are discussed now.

3.2.1 *From Words to Word-Embeddings*

First, there is the problem of how to represent words; the smallest meaningful token in language. For this, distributional word embeddings are the industry standard in which words are represented in an n-dimensional space in which the position and distance between words can represent the different semantic meanings of words and the relationships between them respectively. These methods rely upon the distributional hypothesis, *words that occur in the same context tend to have similar meanings*, and are trained in an unsupervised matter. Starting with *word2vec*[2] that popularized this approach, and many other adaptations that have been made over the years including *GloVe*[3] and *lexvec*[4].

[2] Mikolov et al. (2013b)

[3] Pennington et al. (2014)

[4] Salle et al. (2016)

To evaluate the quality of the obtained word-embeddings, *intrinsic evaluation* benchmarks and *extrinsic evaluation methods* are used[5]. The former is the most popular method to compare language embedding quality as they are dependent upon datasets which make them easy to compare different language embeddings. Intrinsic evaluation benchmarks compare the similarity scores of word-pairs within the embedding-space of language embeddings to the human-based similarity scores of the same word-pairs. If the similarity between word-pairs within the language-embeddings corresponds with our own judgment, a high intrinsic evaluation benchmark score can be expected.

[5] Jones and Galliers (1995)

On the other hand in extrinsic evaluation tasks, the word-embedding quality are evaluated on a direct down-stream performance task such as the TALL-task which therefore makes the quality of the embeddings dependent upon the usability for a particular task. As no standardised down-stream performance tasks exist, extrinsic evaluation tasks are not frequently used to compare language embedding quality directly. In our evaluation setup we both use intrinsic and extrinsic evaluation methods, as intrinsic evaluation benchmark scores are generally faster to calculate and it is widely considered that extrinsic and intrinsic evaluation benchmark scores are highly correlated. Although recently questions have been raised whether this assumption is indeed correct and not highly dependent upon the actual task (Section 3.2.2). Now follows a more detailed explanation of the different intrinsic evaluation categories.

3.2.2 *Intrinsic Evaluation Methods*

Human-based similarity tasks can be divided into categorization-, similarity- and analogy- based tasks that test different aspects of the quality of word-embeddings relying on human-based similarity measurements. Recently Bakarov (2018) provided an overview of the specific ways DSM can be evaluated. For completeness, the purpose of these tasks is mentioned below as these will be later used in our evaluation setup.

However, it should be mentioned that concerns regarding intrinsic evaluation methods have recently been raised[6,7,8]. Specifically, whether the evaluation of word-embeddings quality should not be shifted from intrinsic evaluation benchmarks towards extrinsic evaluation benchmarks. As for most use-cases the amount of training data is limited and word-embeddings are mostly being used with minimal fine-tuning, Jastrzebski et al. (2017) argue that a quantitative measurement is needed to indicate the ease in which knowledge can be transferred to a particular task of interest (see Figure 3.4). In addition, they argue that additional performance metrics are required that indicate the ability of language embeddings to perform well under tasks that require increasingly more non-linear models which make the transfer of knowledge increasing more difficult.

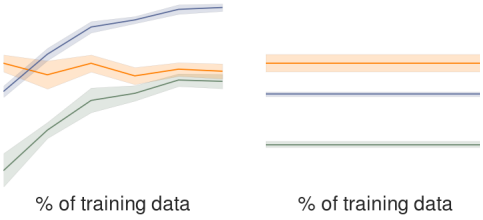


Figure 3.4: Differences in performance obtained on the SimLex intrinsic evaluation benchmark of NMT (yellow), GloVe_300 (blue) and HPCA (green) language embeddings under different percentages of the training data. On the vertical axis is the performance on the task as discussed in the work of Jastrzebski et al. (2017). In supervised-versions of the benchmark (left) one can see that GloVe outperforms the other language embeddings with increased dataset sizes while in the unsupervised version (right) this is not the case. This is used as an argument for using a quantitative metric to denote the ease of KT as an evaluation criterion for language embeddings. Figure reproduced from Jastrzebski et al. (2017).

[6] Schnabel et al. (2015)
[7] Faruqui et al. (2016)
[8] Jastrzebski et al. (2017)

Categorization-based tasks include; BLESS[9], AP[10], BATTIG[11] and ESSLI[12].

In categorisation tasks, the objective is to evaluate the word-embeddings quality regarding their ability to accurately capture the semantic groups they are located in. For example, whether a variety of animals are clustered tightly together in semantic space. One example of the inner workings of these methods is now given.

BLESS consists of 200 concepts containing single-word nouns with relationships to other words. These relationships include COORD (hyponym), HYPER (hypernym), MERO (part of), ATTRi (attribute of), EVENT (an event the concept is involved in) or RAN (random pairing similarity score). The cosine similarity scores between these word-pairs are calculated to test to which extent these embeddings capture these relationships. A high cosine-similarity is desirable for all word-pairs in the different groups except RAN, which behaves like a control group that should have low cosine similarity scores as this group consists of fixed random word-pairs.

[9] Baroni and Lenci (2011)
[10] Almuhareb (2006)
[11] Baroni and Lenci (2010)
[12] Bullinaria (2008)

Similarity-based tasks include; MEN[13], MTurk[14], RG65[15], RW[16], SimLex[17], WS353, WS353R and WS353S[18].

These datasets rely on human heuristic judgments of the actual semantic distances between words. Bakarov (2018) provide an overview of the advancements of the similarity-based

[13] Bruni et al. (2014)
[14] Radinsky et al. (2011)
[15] Rubenstein and Goodenough (1965)
[16] Luong et al. (2013)
[17] Hill et al. (2015)
[18] Finkelstein et al. (2002)

datasets. In these datasets, for instance, the word *cup* and *mug* should be semantically similar which is indicated on a scale between 0 and 1. A human assessor is given a set of pairs and is asked to rate the degree of similarity, for example, the aforementioned pair could have received a similarity score of 0.8. As such, these methods towards obtaining a similarity score have been criticised for their subjective nature, but still are the most popularly used intrinsic evaluation metrics. In specific, what defines *semantic* similarity between words can according to Gladkova and Drozd (2016) be based on up to 50 potential linguistic, psychological and social factors.

ANALOGY-based tasks include; Google[19], MSR[20], SemEval[21].

In an analogy task, the objective is to test whether the obtained semantic embeddings and the relationships between words contain certain relations that allow for arithmetic operations. The most famous example of arithmetic operations come from the Word2Vec paper created by Mikolov et al. (2013c) in which the example is given $vector("King") - vector("Man") + vector("Woman") = vector("Queen")$. The main criticism here is that there is no precise evaluation metric on how to measure this[22].

An example is the SemEval task in which the objective is to determine the degree to which the semantic relationships between word pairs (e.g. A:B, C:D) are similar to each other. Humans were asked to rate the similarity between relationships between two words-pairs. These scores were subsequently used to test whether this similarity was also observed between word-pairs in word-embeddings as a measurement for word-embedding quality.

[19] Mikolov et al. (2013a)
[20] Mikolov et al. (2013c)
[21] Jurgens et al. (2012)

[22] Bakarov (2018)

Question 2: Consider the following word pairs: pilgrim:shrine, hunter:quarry, assassin:victim, climber:peak. These X:Y pairs share a relation, "X R Y". Now consider the following word pairs:

(1) pig:mud
(2) politician:votes
(3) dog:bone
(4) bird:worm

Which of the above numbered word pairs is the MOST illustrative example of the same relation "X R Y"?

Which of the above numbered word pairs is the LEAST illustrative example of the same relation "X R Y"?

Figure 3.5: An example of the method used by Jurgens et al. (2012) to construct the dataset for the anology task; *SemEval*. Based on the human-obtained answers the extend to which the language embeddings are consistent with these findings are used as a measurement for success.

3.3 GraphSAGE

Hamilton et al. (2017) introduced a method called GraphSAGE which stands for SAmple and aggreGatE, that relies upon inductive representation learning on large graphs by learning a function that generates embeddings by sampling and aggregating features from a node's local neighbourhood. The inductive nature of this method allows generalising to unseen nodes or even complete graphs during testing, which is not possible in a transductive approach. However, in order for an inductive framework to work properly, both the node's local and global role in the graph needs to be encoded. This is a computationally challenging task and requires a variety of sampling methods for speed-up[23].

In Figure 3.6 a global overview of the GraphSAGE method is shown. First (1), only a sample of the edges is taken to represent the local neighbourhood (light red) of a particular node of interest (dark red). Second (2), a chosen aggregator function aggregates information for up to n hops away. In the shown

[23] Hamilton et al. (2017)

figure the aggregation of information across two hops is shown which are both learned by a parameterised model (e.g. LSTM) where each aggregator function at a distance n is learned separately but is shared across all nodes. Lastly (3), with this aggregated feature representation both the reconstruction of the local neighbourhood is attempted to be maximised which can include an added node feature representation which in our case are language embeddings. At step (3) one can observe that the obtained feature-representation (the label) of the node should include information about the entire graph. It should be noted that for the GraphSAGE algorithm edges do not have a direction, and instead the arrows in Figure 3.6 refer to the direction of the discussed operations in step (1) and (2).

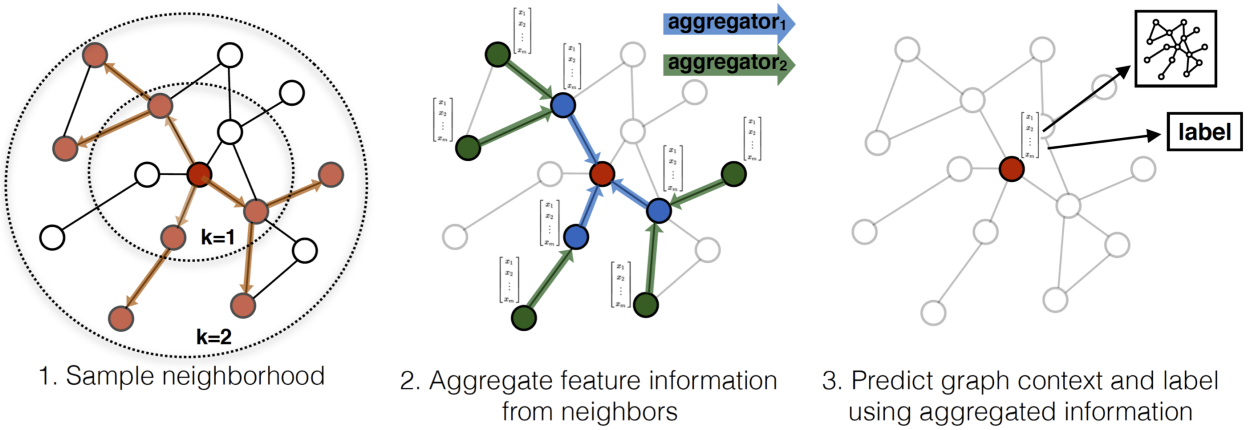


Figure 3.6: Illustration of how the local neighbourhood of a node is sampled (1) after which the node feature representation is aggregated (2) and an unsupervised loss is applied in (3) that attempts to reconstruct the local neighbourhood. Figure reproduced from Hamilton et al. (2017).

The method of Hamilton et al. (2017) was validated on feature-rich graphs including; citation graphs, Reddit and Protein to Protein Interaction (PPI), which contain a relatively high amount of links per node (the average node degree) compared to the average KBs. As a result, ConceptNet which is further discussed in Section 5.2.1, is different in the sense that the average node degree is significantly lower and arguably the direction of the relationships are of importance. For example the edge-relationship type "owned by" is non-symmetric. An example of the difference between directional and undirectional edges is shown in Figure 3.8 and 3.9 respectively. An additional difference is that ConceptNet is significantly larger than the aforementioned graphs.

The connectivity sparsity of ConceptNet could potentially be offset by the fact that nodes can be given additional node feature-representations in the GraphSAGE algorithm which allows the possibility to add distributional word-embedding node-feature representations. This could potentially uplift part of the connection sparsity of ConceptNet as the feature-rich word-embeddings allows relating all words (≈nodes in ConceptNet) in a 300-dimensional space. Also, Kipf and Welling

(2016) showed earlier that only a limited speed-up was observed going from CPU to GPU for GC-based approaches, possibly allowing large networks to be trained on CPU without significantly increased training time.

Hamilton et al. (2017) extend GCNs to the task of inductive unsupervised graph learning and propose a framework that generalises the GC approach to use trainable aggregation functions beyond simple convolutions. In Algorithm 1 one can observe the pseudo-code in which the embeddings z are obtained for each node which represents a word in the English vocabulary. Therefore this is considered the forward propagation as it is not updating the parameters of the model yet.

Algorithm 1 GraphSAGE embedding generation (i.e. forward propagation) algorithm

Input : Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$; input features $\{\mathbf{x}_v, \forall k \in \mathcal{V}\}$; depth K; weight matrices $\mathbf{W^k}, \forall k \in \{1, \ldots, K\}$; non-linearity σ; differentiable aggregators functions AGGREGATE_k, $\forall k \in \{1, \ldots, K\}$; neighborhood function $\mathcal{N} : v \rightarrow 2^V$

Output : Vector representations $\mathbf{z}_v$ for all $v \in \mathcal{V}$

1: $\mathbf{h}_v^0 \leftarrow \mathbf{x}_v, \forall v \in \mathcal{V}$
2: **for** $k = 1 \ldots K$ **do**
3: **for** $v \in \mathcal{V}$ **do**
4: $\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow AGGREGATE_k(\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\})$
5: $\mathbf{h}_v^k \leftarrow \sigma(\mathbf{W}^k \cdot CONCAT(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k))$
6: $\mathbf{h}_v^k \leftarrow \mathbf{h}_v^k / ||\mathbf{h}_v^k||_2, \forall v \in \mathcal{V}$
7: $\mathbf{z}_v \leftarrow \mathbf{h}_v^K, \forall v \in \mathcal{V}$

A step-by-step explanation of the feed-forward algorithm is given below with the line number indicated between brackets. The input of the model is a graph $\mathcal{G}$ consisting of vertices $\mathcal{V}$ and edges $\mathcal{E}$, together with a feature representation of $\mathbf{x}_v \forall v \in \mathcal{V}$. First, at the start of the algorithm (1) the node feature representations are taken as the original representation of v called h. One can interpret k as the time-step or number of hops the representation of node v is dependent on. At time-step 0, the representation therefore only consists of the node-features and no local neighbourhood information is incorporated yet. Thereafter, for search depth K, the local neighbourhood of the node of interest v is aggregated with increasing depths (4). This is accomplished by for example taking the *average*-aggregator function that takes the mean of the neighbourhood region. Thereafter the initial and the current feature-representation are concatenated and weighted by W that is different per k after which a non-linearity function is applied (5). This can be considered a skip-connection between the current and previ-

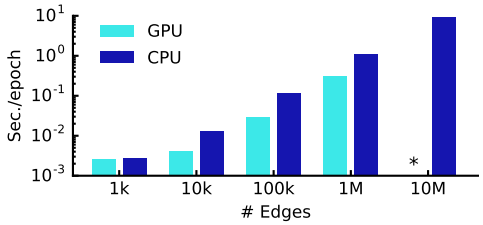


Figure 3.7: A near linear increase in computational time is observed when the number of edges increases with minimal difference between GPU and CPU time. Figure reproduced from Kipf and Welling (2016).

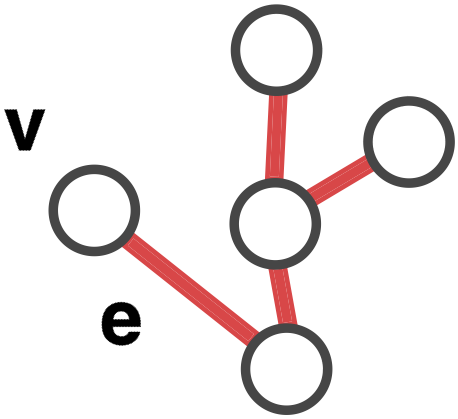


Figure 3.8: Illustration of a graph where all the edges are of the same relation. From a modeling perspective, Hamilton et al. (2017) considers all edge-types equal and without directionality.

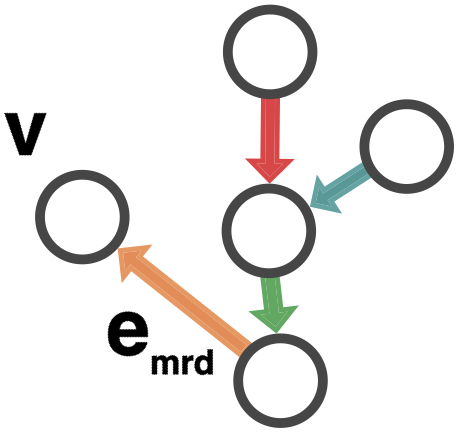


Figure 3.9: Illustration of a graph where there are different edge-relations of which the directionality is important. Many graphs belong into this category, including ConceptNet and ImageNet.

ous hop. Lastly, the feature representations of the nodes are individually $l2$-normalised at each time step k (6) that at the last time-step K results in the latent feature representation z_v.

To guaranty that each mini-batch has the same memory requirements to allow for faster training, the neighbourhood function $\mathcal{N}$ was capped at a user-defined parameter S. Therefore the time complexity of this algorithm is $\mathcal{O}(\prod_{i=1}^{K} S_i)$, where S_i denotes the maximum number of connections allowed at search-depth i. Hamilton et al. (2017) empirically found that the parameter-settings $K = 2$ and $S_1 \cdot S_2 \leq 500$ performed best for the datasets that were used.

3.3.1 *Training*

An unsupervised loss-function was used that enforces closely connected nodes to have similar representations while disparate nodes have highly distinct representations. For this, the output representations of the nodes obtained at the end of Algorithm 1 was used with trainable parameters W^k. For positive examples, a random node v is sampled in the neighbourhood of n edges apart of v while for the negative examples Q examples are sampled from a distribution called P_n which is the negation of the previous set,

$$J_{\mathcal{G}}(\mathbf{z}_u) = -\log(\sigma(\mathbf{z}_u^T \mathbf{z}_v)) - Q \cdot \mathop{\mathbb{E}}_{v_n \sim P_n(v)} \log(\sigma(-\mathbf{z}_u^T \mathbf{z}_{v_n})) \quad (3.7)$$

As the latent node representations $\mathbf{z}$ have been l2 normalised, the dot product between either similar (v within the neighbourhood) nodes or dissimilar nodes (v outside neighbourhood) are 1 and -1 under ideal circumstances respectively. The sigmoid function after that normalises these values between 0 and 1 after which a log is applied for numerical stability.

3.3.2 *Aggregators*

GraphSAGE provides four different aggregator functions, GC, mean, LSTM, pool, which all obtain relatively similar performance on their datasets[24]. The differences in running time of these aggregators can be seen in Figure 3.10.

Mean Aggregator The mean aggregator function is formalised as;

$$\mathbf{h}_v^k \leftarrow \sigma(\mathbf{W} \cdot MEAN(\{\mathbf{h}_v^{k-1}\} \cup \{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\})) \quad (3.8)$$

As a result, the pseudocode on line 5 in Algorithm 1 is altered and instead of applying the concatenation of node averages, the node feature representations are averaged.

LSTM Aggregator The LSTM network architecture can express more information than the mean aggregator function

[24] Hamilton et al. (2017)

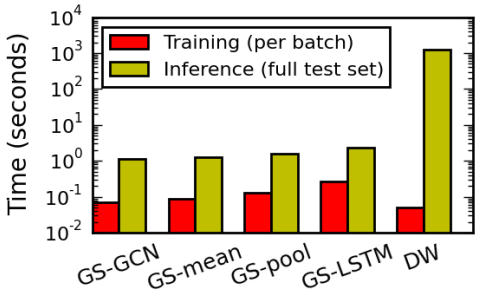


Figure 3.10: An overview of the training- and testing-time of the different aggregator functions. DW stand for DeepWalk which is one of the benchmark methods used by Hamilton et al. for comparison. Figure reproduced from Hamilton et al. (2017).

potentially can. However, due to the sequential nature of LSTM's which are not in accordance with the unordered neighbourhood, Hamilton et al. adapt LSTMs to work on a random permutation of the node's neighbourhood.

Pooling Aggregator This symmetric and trainable aggregator function feeds each neighbour's vector independently through a fully connected neural network after which an element-wise max-pooling operation is applied to aggregate the information.

$$AGGREGATE_k^{pool} = max(\sigma(\mathbf{W}_{pool}\mathbf{h}_{u_i}^k + \mathbf{b}), \forall u_i \in \mathcal{N}(v) \quad (3.9)$$

For more details on their code implementation of the aforementioned aggregator functions we defer to their working implementation on github.

3·3·3 *Sampling*

As the distribution of edges per node is highly skewed, Hamilton et al. sample only some edges for each node before feeding them into the GraphSAGE algorithm. In particular, a maximum degree of 128 per edge was sampled pre-training while sampling only 25 neighbours of those during training. The downsampling of edges allows the storage of the node neighbourhood information as dense adjacency lists pre-training, which drastically improves computational efficiency during training as minimal lookup time is required during training.

3·4 *Zero-shot Learning*

In our introduction, we hypothesised that a high zero-shot performance was beneficial for the task of event-localisation given natural language text. In the upcoming section, we formalise zero-shot learning and end with recommendations of Xian et al. (2017) towards better zero-shot evaluation practices.

In zero-shot learning the objective is to learn a classifier $f : X \rightarrow Y$ that can predict unseen classes in Y that were not in the training set. Formally this means that given a training set $\mathcal{S} = (x_n, y_n), n = 1 \dots N$ and $y_n \in \mathcal{Y}^{train}$, the aim is to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ by minimizing a loss-function:

$$\mathcal{L}(y_n, f(x_n; \mathcal{W})) + \lambda(\mathcal{W})$$

where $\lambda(\cdot)$ being a regularization term and

$$f(x; \mathcal{W}) = argmax_{y \in \mathcal{Y}} F(x, y; \mathcal{W})$$

is maximizing the likelihood of y given the model parameters $\mathcal{W}$. What is different from a typical machine learning setting is that during testing unseen classes are given to the model $y_n \in \mathcal{Y}^{test}$ with $Y^{test} \cap Y^{train} = \varnothing$ and the aim is to still correctly

predict the y label for the test-set. Zero-shot learning settings are especially important in cases that (1) $\mathcal{Y}$ can take many values and (2) the cost of obtaining labeled examples is high[25]. The cross-modality learning between language and vision for the localisation of events falls under these criteria due to (1) the language domain contains many words with many complex relationships between them while (2) temporally annotated datasets are rare due to the high labeling cost[26]. Therefore these two factors make event-localisation given natural language text to a large extend a zero-shot learning problem.

[25] Palatucci et al. (2009)

[26] Dai et al. (2017), Ma et al. (2017)

Despite the increased popularity of zero-shot learning, a comparison between zero-shot methods remains difficult due to the absence of unified bench-marking methods[27]. In specific, most methods use their own test and training set split on popular zero-shot evaluation datasets which make direct comparisons difficult. Moreover, zero-shot methods that rely upon pre-trained visual representations frequently violate the $\mathcal{Y}^{test} \cap \mathcal{Y}^{train} = \varnothing$ constraint as they contain classes in the train set that are either close or exactly equal to the classes in the test-set. By replicating the work of recent zero-shot methods with test-set splits of popular evaluation benchmarks that do not contain training set classes, Xian et al. show that some methods achieve significant performance losses while others even benefit from the proposed split further stressing the importance of a proper training and test-set split. This demonstrates that extra care is needed towards constructing training and test-set splits especially when pre-trained visual representations are used on a different task. This also becomes important in Section 4.3 where we create our own general zero-shot dataset based on ImageNet.

[27] Xian et al. (2017)

Arguably, most zero-shot evaluation methods focus on narrow zero-shot performance by focusing on images within a narrow domain and only a few image classes[28] instead of focusing on large-scale zero-shot performance that is presumably beneficial for the TALL-task. Rohrbach et al. (2011) provide a first in-depth study towards knowledge transfer and zero-shot learning in a large-scale setting by focusing on the ImageNet dataset as a method to obtain a wide-scale zero-shot performance score. For the representation of language, the synset definitions originating from WordNet were used for attribute mining to combine the vision and language modality and use these attributes to generalise even to unseen classes. To circumvent the sparsity that occurs when only directly matching overlapping attributes between textual descriptions, instead the *semantic relatedness* between the mined attributes was calculated to improve relating the different attributes and therefore textual descriptions to each other. Rohrbach et al. (2011) state that it remains questionable whether the performance that is obtained using algorithms operating within narrow domains also

[28] e.g. birds - CUB Welinder et al. (2010), scenes - SUN Patterson and Hays (2012), animals - AWA Lampert et al. (2014), objects - aPy Farhadi et al. (2009)

scale to larger number of classes and training data. Rohrbach et al. conclude with showing that by exploiting the hierarchy of the WordNet hierarchy for obtaining semantic relatedness between classes, zero-shot classification performance improved when compared to attribute mining using a direct similarity approach. In our approach the hierarchy of ConceptNet is used in the hope of observing a similar result.

3.4.1 *Zero-Shot Evaluation Metrics*

For the evaluation of zero-shot performance, recently Xian et al. (2017) propose to use the per-class averaged top-1 accuracy if the dataset is not well balanced with respect to the number of images per class and argue that demonstrating zero-shot performance on small or coarse-grained datasets are not recommended. Instead, they recommend to abstract away from the restricted nature of zero-shot evaluation and make the task more practical by also including training classes in the search space, which make it equal to a GZSL task-setting. In addition, Xian et al. (2017) recommend using multiple test-set splits to decrease the amount of variance.

For GZSL it is common practice to compute the harmonic mean of training and test-set accuracies,

$$H = 2 * (acc_{\mathcal{Y}^{tr}} * acc_{\mathcal{Y}^{ts}}) / (acc_{\mathcal{Y}^{tr}} + acc_{\mathcal{Y}^{ts}}) \tag{3.10}$$

In comparison to harmonic mean, the arithmetic mean is more affected by high training set accuracies and is therefore not recommended as the test-set accuracies tend to be considerably lower. As a result in our work we include training classes in our evaluation benchmark as discussed in Section 5.1.3. Instead of reporting the harmonic mean, however, we report both the train and test-set accuracies and leave further calculations to the reader.

4 Methods

Whereas in the Background Section (2) a general overview was provided of the literature that helped shape our view upon the problem of event-localisation in videos and in Related Work the GraphSAGE and TALL model architecture was outlined as they are essential to the work presented here, in this section we provide a general overview of the methodology used in this work.

We start with giving a formal problem-statement of the TALL-task (4.1), after which we provide an outline of our approach (4.2). Thereafter we go in-depth to each of our individual experiments by starting off with their relation to the research questions as formulated in the Introduction (1.4) after which we provide a more detailed overview of the methods. The same structure is adhered to in the following Experimental Setup Section (5) in which the implementation details are discussed.

4.1 Problem Formulation

Given a video V, consisting of T frames $f_{t=1}^{T}$ and temporal sentence annotations $A = \{s_j, \tau_j^s, \tau_j^e\}_{j=1}^{M}$ with M sentences where superscript s and e denote the start and ending time of the event described in text s, the objective is to predict the temporal boundaries (τ^s, τ^e) of a particular event s_j of a given V. The models' input are corresponding sentence and video pairs and the output are the temporal boundaries that correspond to the event the sentence refers to. For our purpose the feature representation of vision and language are fixed.

This is a similar task description as first described in Gao et al. (2017) which called this the TALL-task. Gao et al. mainly focus on obtaining an appropriate model design that allows for learning a cross-modal embedding space $\mathcal{E}$ in which language $\mathcal{T}$ and vision $\mathcal{V}$ can be matched, here we solely attempt to improve the language representation $\mathcal{T}$ inline with our hypothesis to enhance the alignment between the two modalities such that even accurate matching can take place for zero-shot use-cases.

4.2 Overview of Experiments

In this work, an attempt is made to improve upon the representation of language specifically for the TALL down-stream performance task in which natural language text is used. As argued, embeddings for this particular task should hypothetically be (1) more prominently centred around the functional roles objects take part in while also emphasising on relations

that have clear visual correspondences, while (2) also allowing for the transfer of knowledge between the linguistic and visual domain. As argued this makes it close to a general zero-shot task-setting in which high performance on both the train and test-set vocabulary is essential.

To incorporate (1) and take into considerations (2), the relational knowledge available in KBs were combined with distributional semantic approaches. As knowledge bases are incomplete and are frequently having only a few relationships per entity, learning embeddings solely from relational knowledge was assumed to be too sparse. Our approach is built upon two realisations. First, nodes in ConceptNet are represented by *concepts* which can be seen as a link to their distributional embedding word-representation allowing nodes to be enriched with these embeddings. Second, recently unsupervised GCNs have been proposed to create individual node-representations by aggregating the local neighbourhood information into a low dimensional space similar to DSM. Specifically GraphSAGE[1], an inductive framework that leverages node feature information and local neighbourhoods to obtain node embeddings for large knowledge graphs, was considered useful for this particular task.

[1] Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1025–1035

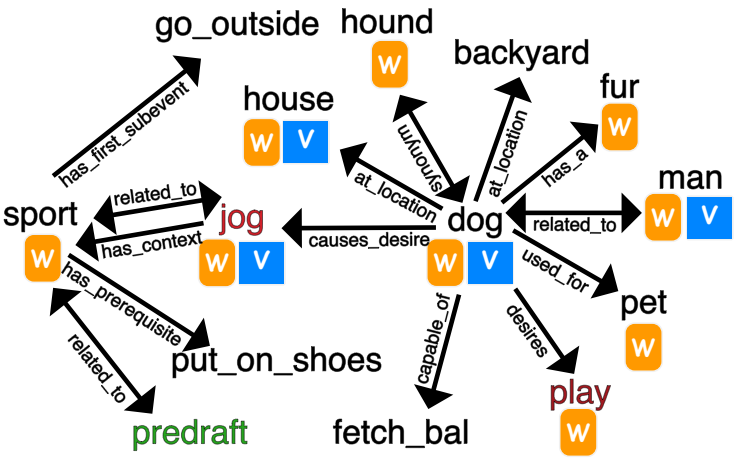


Figure 4.1: An example of how GraphSAGE can be applied upon ConceptNet. W in yellow represents word-embeddings, V represents a visual representation. In red are the verbs and in green are the adjectives. Arrows indicate either undirected (*e.g. related_to*) or directed edges (*used_for*). This figure illustrates that for some concepts there are visual and textual correspondences whereas for others there is not. This possibly allows to relate unrelated to related concepts, ideal for zero-shot use-cases.

The hypothetically more structured embeddings trained with GCNs on a KB could improve the GZSL task performance as it would allow to more systematically transfer knowledge from the known visual-language correspondences during training to unseen visual and linguistic examples during testing. In Figure 4.1 an example is given of the potential of this approach using an actual sub-graph available in ConceptNet. The yellow w indicates that for this particular concept in ConceptNet a corresponding DSM word-embedding representation was found in the vocabulary. This feature node-representation was therefore added to the graph to partially neglect the sparsity of the relationships in ConceptNet. The blue v indicates that

a matching visual image-feature was found for the particular concept of interest which consequently forms a text-visual correspondence that is used to obtain a cross-modal embedding space only after the language embeddings are obtained using GraphSAGE.

During the training of GraphSAGE not all concepts had a matching language representation, while during the training of the cross-modal embedding space not all nodes contained visual correspondences. The structure of the graph can therefore help here in two distinct ways. First, an accurate language representation for concepts without language-representation can still be obtained using GraphSAGE by aggregating information from the local neighbourhood. Second, the more structured language embeddings that were expected to be obtained using this approach could hypothetically be used to better align vision and text even for ZSL use cases. For example, only knowing which relationships a concept has can already tell a lot about its functionality, which is frequently visually reflected by its shape or form.

The relationships in ConceptNet and the frequency thereof are shown in Figure 4.2. Here one can observe that a significant part of relations is related to hypernyms and hyponyms (isa, partof, formof), synonyms and antonyms, or functional roles (e.g. usedfor, atlocation, hassubevent, capableof, hasproperty). These relations could be important for the creation of language embeddings specifically for the task of event-localisation as they are centred around function with often clear visual correspondences. Knowledge transfer can also be enhanced by for example knowing that a *Chihuahua* dog is a more fine-grained example of the class *dog*. Given that visual-linguistic correspondences are known for the class dog in, for example, ImageNet could be exploited by requiring that their representations must be close to each other in semantic space.

Our approach can be subdivided in three experiments. The first experiment is designed to test to which extend popular language embeddings are already suited to be used to train a cross-modal embedding space in which seen and unseen language and visual correspondences can be matched during test-time in a GZSL-setting. For this, a new zero-shot dataset is created that exploits the ImageNet hierarchy to enforce that the synsets in the train and test-set are sufficiently distinct while also benefiting from the wide variety of objects and number of image-examples available in ImageNet. The textual descriptions of the synset (.e.g *dog*) are subsequently matched with the vocabulary of the language embeddings after which a model is trained that projects the visual-representation close to their respective language-representation. The obtained cross-modal embedding space is subsequently used to evaluate the model's ability to give higher similarity scores to matching visual and

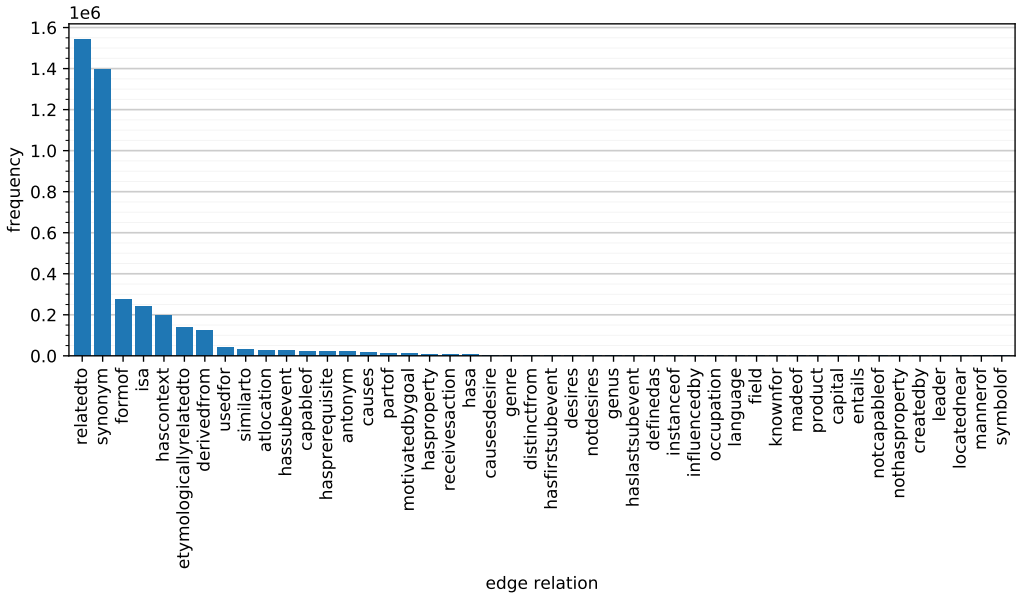


Figure 4.2: Frequency of edge-relationships in ConceptNet. Based on the final selection of concepts from ConceptNet.

text correspondences. The model's ability to rank the similarity of corresponding visual and textual representations higher than non-corresponding ones is used as a measurement for zero-shot performance. This zero-shot performance metric is then later compared to the downstream performance task in Experiment III in order to determine whether there is a positive correlation between the two (RQ3).

In the second experiment, embeddings are trained specifically with our hypothesis in mind that embeddings trained with an emphasis on zero-shot use-cases and visually grounded relationships perform better in the TALL-task. For this, ConceptNet is adapted to fit the input-structure of GraphSAGE and language-embeddings are added to each concept which requires a similar matching method as was used in Experiment I. Heuristics are used to even obtain language representation for words that do not occur in the vocabulary of the distributional word-embeddings, resulting in each concept in ConceptNet having a corresponding DSM word-embedding representation. Experiment one is then repeated for the language embedding obtained using GraphSAGE to compare the zero-shot performance of these embeddings with others and later with the results obtained in Experiment III. The hyper-parameters of Graph-SAGE are fine-tuned on intrinsic evaluation benchmarks as it is widely accepted that intrinsic evaluation benchmark performance is a decent indicator of extrinsic evaluation benchmark performance. The main benefit of this approach is the significantly faster evaluation time of intrinsic evaluation benchmarks. The performance on the actual extrinsic evaluation task, the TALL-task, is calculated in Experiment III.

The third experiment is used to compare the performance

of the language representations in the actual downstream performance task using the model architecture and dataset introduced by Gao et al. (2017). Experiment one and three can then be used to validate our hypothesis that higher zero-shot is expected to result in better performance in the TALL-task. As the former model architecture takes in language representations on the sentence-level rather than the word-level, different methods are compared to go from word- to sentence-representations. Lastly, to test the extent the approach of Gao et al. (2017) actually relies on an accurate representation of language, the vocabulary of the training set is one-hot encoded after which a sentence embedding is created out of the average of a sentence's word-representations. As Gao et al. (2017) introduced the TALL-task to distance itself from the one-hot encoding of only a select number of event-classes, this was deemed necessary to validate whether their used model and evaluation benchmark actually required a richer representation of language that goes beyond direct vocabulary/class matching.

In the Background Section (2) we have provided an overview of literature that showed that datasets within the video domain are still significantly smaller and less diverse than for example ImageNet. In addition, working within the video domain increases the computational cost and difficulty of finding an appropriate visual representation. To partially circumvent these issues and focus on the main objective here which is to obtain an improved representation of language for event-localisation, Experiment I was conducted in the image domain. This allowed for quicker experimentation while working with larger and more diverse datasets, which we expected to better correspond to the general nature of events in the TALL-task.

4.3 *I - Zero-shot Cross-Modal Embedding Space Evaluation*

4.3.1 *Objective & Relations to Research Questions*

In the introduction, we hypothesised (H1, Section 1.5) that more structured language embeddings were beneficial for better performance in the TALL-task due to improved zero-shot performance in the cross-modal embedding space for unseen visual-linguistic correspondences. To examine to which extend the currently popular semantic word embeddings could be aligned with visual representations for even unseen classes, this experiment was conducted that focused on the zero-shot capabilities of the different language embeddings when being matched with their visual counterpart (e.g. a textual and visual feature representation of a "dog"). The result of this experiment is a zero-shot evaluation dataset that allows us to obtain a quantitative measure to which extent language embeddings can be matched with visual features for seen and unseen classes.

In RQ1 we pose the question whether the combination of both distributional and relational knowledge is beneficial for aligning both modalities and therefore results in higher zero-shot performance. As in Experiment II we obtain our own language embeddings using both distributional and relational knowledge similarly to Numberbatch, this allows us to compare whether the addition of relational knowledge is beneficial for improved zero-shot performance. In RQ3 we compare the quantitative zero-shot evaluation metric obtained in this experiment, Experiment I, with the actual performance on the TALL-task obtained in Experiment III. The difference between Experiment I and Experiment III is that in the former we test our hypothesis that adding relation knowledge improves zero-shot performance whereas in the latter we actually test whether this leads to the hypothesised improvements on the TALL-task.

4.3.2 *Methods Overview*

To evaluate the zero-shot performance of a variety of language embeddings when being matched with visual cues, it is beneficiary to create a dataset in which similar classes in the train and test-set are ruled out to further stress the importance of broad rather than narrow zero-shot performance. This is in accordance with the wide variety of events in the down-stream performance task; TALL. ImageNet, a hierarchical database, allows for this due to the WordNet structure that expresses the object-classes in a parent-child hierarchy. Consequently, this information can be used to ensure that children of parent classes are excluded in the test-set as they tend to have a high visual-similarity with the parent class which makes it an easier challenge. In addition, in comparison to most data sets within the video domain, ImageNet is considerably more diverse and contains a general set of objects/entities[2]. Xian et al. (2017) recently released specific zero-shot splits of ImageNet using a similar approach as proposed here. As the splits were unreleased during the creation of our test-set, we have proposed three test-set splits of our own named; *narrow*, *internal*, *random*. It should be noted, however, that ImageNet only contains low-level events such as objects and lack any mid- to high-level events such as shooting a ball or playing football. Arguably, mid- and high-level events can be best captured within the video- rather than image-domain, as they frequently require complex motion patterns to be captured. This is left for future work.

[2] Abu-El-Haija et al. (2016)

For these datasets, visual features were extracted using the pre-trained Inception-V1 architecture after which a network was trained that projected these features as close as possible to the linguistic representation of the synset. Each synset in ImageNet has a Unique Identifier (UID) which has a corresponding textual description that describes the synset in

words. These descriptions can be used to match them with the vocabulary of word-embedding methods in order to obtain a feature-representation of the synset in language. With the feature-representations of images extracted from a pre-trained CNN in a synset and corresponding word-representation in place, we use this to train a cross-modal embedding space in which both the visual and textual representations need to be close. This is used as a simplified version of the cross-modal embedding space needed to perform the TALL-task in Experiment III (4.5). The benefits of conducting this experiment is that the zero-shot performance is tested in isolation of the interactions with other components (e.g. temporal localisation) and faster training and evaluation times are obtained by working within the domain of images when compared to the TALL-task.

For each word-embedding that was selected, *Glove* (glove8 40B 300d.txt), *Word2Vec* (GoogleNews vectors negative300), *LexVec* (lexvec commoncrawl 300d W-pos vectors), *Numberbatch* (numerbatch en 17.06) and in later experiments also the language embedding obtained by us, a projection-network is trained separately that projects the visual features into the domain of the language embeddings. The decision was made not project both modalities into a new space or project the language features into the visual domain for model simplicity and the fact that our visual features are of higher dimensionality (1024 compared to 300) respectively. The unseen classes from the test-set are then projected in a similar fashion. The word-embeddings were selected by popularity and intrinsic evaluation scores (see Table 6.2). Within the cross-modal embedding space, the similarity between each visual-synset (all feature representations of the images in a particular synset) is calculated to all the language representations of all synsets. Ordered on their similarity, this results in a ranking score in which ideally the projected visual representation V'_j and corresponding textual representation T_j has the highest similarity of all word-embeddings T. The mean Average Precision (mAP) metric is used as an indication of the cross-modal embedding space to even generalise to unseen classes. To be in line with the harmonic mean performance metric which is commonly used in GZSL-tasks, both the performance on seen and unseen classes are reported.

For zero-shot purposes, having internal nodes in the training-set of which the children are in the test-set could lead to an over-estimation of performance as these children tend to be generally close to the parent node. An example is demonstrated in Figure 4.3 where one can observe that there is a significant overlap between visual features between synset children-classes. The result is that these synsets have similar feature representations through the considerable overlap in the model's input-space, the images, which arguably makes it less

of a zero-shot performance task. Therefore an extra effort was made to select synsets for the test-sets specifically with respect to their relative position to the synsets in the training set using the ImageNet synset-hierarchy.

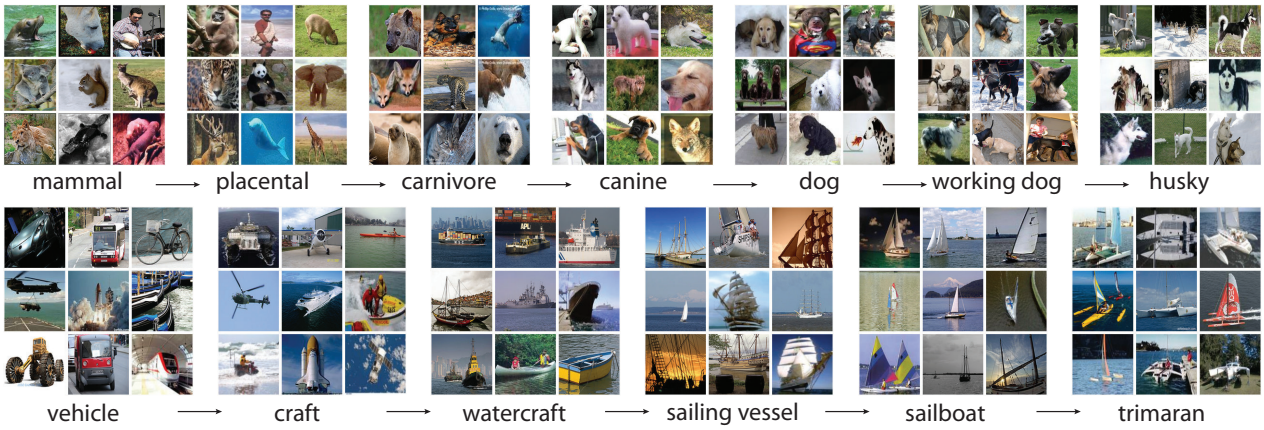


Figure 4.3: A visual example of how the synsets in ImageNet share visual correspondences between more specific examples in the ImageNet-hierarchy. The arrows indicate *is_a* relationships with the more general parent class being on the left-side. Figure reproduced from Deng et al. (2009).

4.4 II - GraphSAGE-ConceptNet Embeddings

4.4.1 Objective & Relations to Research Questions

In this experiment, GraphSAGE is applied on ConceptNet with additional DSM node feature embeddings to obtain word-embeddings that combine relational knowledge with distributional word semantics. It was expected that the structure of ConceptNet could be used to enhance zero-shot performance by allowing knowledge to be transferred from corresponding vision-language pairs during training to unseen examples. In addition, the many relationships in ConceptNet centred around objects, and the role they play in relation to other concepts was expected to result into embeddings that could be better aligned with visual representations. However, these assumptions need testing. First, whether our approach indeed leads to language embeddings that contain these properties. In Experiment I we test whether our language embeddings indeed improve zero-shot performance when matching linguistic with visual features. Second, testing is required whether these properties actually lead to the hypothesised improved results for the TALL-task. As previously mentioned, this is tested in Experiment III. Therefore Experiment II is used to incorporate our hypothesis into a potential solution which further tested in Experiment I and Experiment II to answer RQ1 and RQ2.

4.4.2 Methods Overview

To create our own language embeddings using GraphSAGE on ConceptNet with node-embeddings features, a couple of challenges had to be faced.

First, GraphSAGE was successfully applied to datasets in

a different domain which poses the question of whether this method can be used on ConceptNet. ConceptNet was found to be both larger in scale and had significantly lower connectivity. To ensure that the training of GraphSAGE yielded embeddings of quality for our task, 17 different intrinsic word-embedding evaluation benchmarks were run after model-convergence as a time-efficient alternative to running the TALL-task evaluation benchmark which requires training the model first. Despite the recent controversy of using intrinsic evaluation benchmarks as an indication of extrinsic evaluation benchmark success[3,4,5], this still remains common practice in literature. To allow GraphSAGE to be run on the structure of ConceptNet, first a selection was made that focused only on the English vocabulary and removed disconnected sub-graphs that were deemed irrelevant for our task to greatly reduce the original data-set size. Subsequently, the parameters of GraphSAGE were adapted to our domain including; the amount of sampling and amount of hops allowed to aggregate the local neighbourhood information. For this, both the memory requirements of the model and the performance on intrinsic evaluation metrics were taken into account.

[3] Chiu et al. (2016)
[4] Faruqui et al. (2016)
[5] Jastrzebski et al. (2017)

The second challenge is to match the ConceptNet nodes, which are represented by words (*e.g. "dog"*), with a corresponding language representation. This is not trivial as the vocabulary of ConceptNet, the concepts, do to a large extent not overlap with the vocabulary of DSM methods. In our approach, we solved this mismatch of the ConceptNet concept-vocabulary and DSM vocabulary by choosing Numberbatch which was found to have the highest vocabulary-overlap while replacing unknown embeddings with values obtained by heuristics. Three different replacement techniques were used; taking the local neighbourhood average, using zero-vectors or an average of all word-vectors.

Lastly, a significant amount of time was dedicated towards the hyper-parameter tuning of GraphSAGE to find out how the different aggregator functions and parameters affected the obtained node-feature representations (word-embeddings). Ultimately this resulted in two-aggregator methods being selected which were obtaining similar performance but using fundamentally different methods. The language embeddings obtained in this experiment were then compared to selected popular language embeddings based on their performance on the TALL-task (in Experiment III) and our obtained zero-shot dataset (in Experiment I).

4.5 III - TALL with Sentence Embedding Replacements

4.5.1 Objective & Relations to Research Questions

In this experiment the objective is to relate the zero-shot evaluation scores obtained in Experiment I of popular language-embeddings and our own (obtained in Experiment II) to the actual performance of these language embedding in the TALL-task. As aforementioned, this allows us to give a potential answer to RQ3. This was assumed to be a reliable method to test whether more structured language embeddings would be beneficial for event-localisation given natural language text.

4.5.2 Methods Overview

In order to make a direct comparison between the ability of the different language embeddings to be aligned with visual features in the TALL-task, the model architecture and evaluation setup of Gao et al. (2017) was used with the representation of language being substituted by the language embeddings used in Experiment I and the ones that were obtained in Experiment II. Gao et al. (2017) use the TACoS and Charades-STA dataset for evaluating the performance on the TALL-task and provide access to the pre-processed and sentence annotated TACoS dataset here. As the Charades-STA dataset is not made publicly available and the mining of training-examples is not straightforward (Section 3.1.2), this dataset was not used. To compare the performance of the *Word2Vec*, *LexVec*, *Numberbatch*, *Glove* and our own language embeddings, we replaced the sentence annotations in the provided TACoS dataset. Skip-thought[6] was originally used to represent sentence feature-representations of textual queries within the TACoS and Charades-STA datasets. This sentence representation is trained on the sentence- rather than the word-level. To also obtain sentence embeddings using our word-level language representation in order to substitute their Skip-thought language representation, two approaches were taken.

First, to obtain a sentence representation from the word-embeddings the average of the individual words in the sentence were taken to obtain a 300-dimensional sentence representation. This allowed for direct comparisons between the different word-embedding methods. Second, InferSent was used as a SOTA unsupervised technique to obtain 4800-dimensional sentence embeddings from pre-trained word-embeddings. Thereafter the TACOS-dataset was updated with these different language representations and the method of Gao et al. was repeated under default settings to compare the ability of the model to localise the sentence representations in their test-set. No stop-words were removed as there was no mentioning of this in the original paper.

[6] Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302

Lastly, as the main contribution of the TALL-task is to allow the use-age of natural language text instead of a one-hot-encoding of event-classes, an attempt was made whether their combination of method and dataset actually requires the need for such a representation for high performance. For instance, a completely overlapping and small vocabulary between the train and test-set would be relatively close to the original task of event-classification or event-localisation given only a small list of one-hot-encoded event classes. According to our view, the task of using natural language text is much more similar to a GZSL setting, where the objective is to relate the limited textual-visual correspondences during training-time to unseen correspondences during test-time. Therefore to validate their (evaluation-)method, it was deemed necessary to first test to which extent knowledge transfer between unseen and seen vocabulary was required to perform well in their designed evaluation benchmark. To accomplish this, the overlap between the train and test-set vocabulary in the TACoS dataset was calculated as well as the vocabulary size. The words were then one-hot encoded and a sentence embedding was obtained by averaging the word-vector representations. If with this language representation still high performance on the TALL-task could be obtained this would be a clear indication that this evaluation setup does not emphasise the usage of natural language text.

5 Experimental Setup

In this section, the practical implementation details of the methods as discussed in Method Section (4) are discussed. The experiments and their details are discussed in the same order with matching section titles.

5.1 I - Zero-Shot Evaluation of Cross-Modal Embedding Space

5.1.1 Dataset Creation Details

Deng et al. (2010) introduced a subset of ImageNet synsets consisting of 10184 classes that was used here as a starting point to create our own zero-shot dataset. The dataset is available on ImageNet.org and contains 9114552 images with a minimum of 200 images per class and 800GB in size. To not have a bias towards classes that contain more images, for each class only 200 images were sampled resulting in a total of 2036800 images. Thereafter a pre-trained Inception-V1 network trained on ImageNet1k was used to extract features from to circumvent the need to train the model ourselves (available at TF-slim). As this network was pre-trained on 1000 ImageNet classes, these specific synsets were used as our training-set and acted as a starting point to obtain our test-sets.

The Inception-V1 model architecture was selected because Carreira and Zisserman (2017) recently demonstrated that inflating the relatively old Inception-V1 (2015) network architecture to 3D resulted in SOTA visual feature representations when trained on the Kinetics dataset while being only later fine-tuned on evaluation benchmark datasets. The main reason for selecting this older architecture was the computational efficiency obtained by using the Inception-module that efficiently shares parameters (Table 5.1). As our zero-shot evaluation benchmark is created within the image rather than the video domain to allow the usage of ImageNet, the decision was made to use this model architecture because of the already proven success of inflating this particular architecture to the video domain. As argued, the wide variety of subjects that events can cover arguably demands this large variety of topics which is not yet achievable within the video domain.

To demonstrate the effect that the selection of synsets has on the zero-shot performance, three zero-shot test-sets were introduced that varied in the amount of children synsets of the training-synsets; *random*, *internal* and *narrow*. Each test-set initially contained 1137 synsets before certain synsets were filtered out. For a simplified overview of the differences see Figure 5.1. In *random* (R) the only limitation is that the train-

Network	Layers	Top-1 error	Top-5 error	Speed (ms)	Citation
AlexNet	8	42.90	19.80	14.56	Krizhevski et al.
Inception-V1	22	30.24	10.07	39.14	Szegedy et al.
VGG-16	16	27.00	8.80	128.62	Simonyan et al.
VGG-19	19	27.30	9.00	147.32	Simonyan et al.
ResNet-18	18	30.43	10.76	31.54	He et al.
ResNet-34	34	26.73	8.74	51.59	He et al.
ResNet-50	34	26.73	8.74	51.59	He et al.
ResNet-101	101	22.44	6.21	156.44	He et al.
ResNet-152	152	22.16	6.16	217.91	He et al.
ResNet-200	200	21.66	5.79	296.51	He et al.

Table 5.1: Accuracy and feed-forward + backward time for popular modals using the Pascal Titan X GPU architecture. Benchmarks partially taken from github.com/jcjohnson/cnn-benchmarks.

and test-classes are non-overlapping. In *internal* (I) purposefully only children-synsets are used based on the training-set synsets (X), which are therefore more specific visual examples of the ones available in the training datasets. This test-set is called *internal* as the nodes are within the parent's sub-tree. The *narrow* (N) test-set uses only leaf-nodes that do not exist within the hyponym sub-tree of any of the synsets in the training-set. Given that the children classes share more similarity in both the textual and visual domain, it can be expected that the most difficult zero-shot evaluation test-sets are therefore from the hardest to easiest; *narrow*, *random* and *internal*. This is further discussed in the Result Section (6).

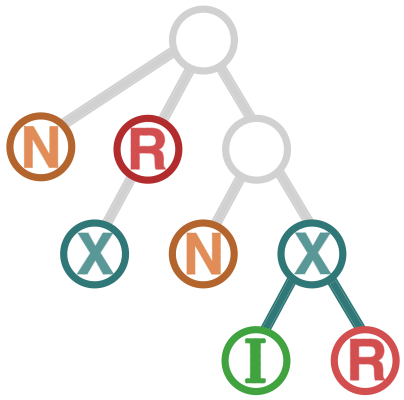


Figure 5.1: An abstract example of how the nodes were selected from the hierarchy of ImageNet. The N stands for nodes that could be selected in the *narrow* dataset. The R stands for *random* nodes, while I stands for *internal* nodes, whereas X represent the nodes in the training-set.

The structure of ImageNet was obtained by parsing the *fall2011 XML* version of the ImageNet (*structure_released.xml*) dataset available on the official ImageNet website. The result of this parse was the hierarchy of all the *is_a* relationships between all synsets existing in the fall-2011 version of ImageNet. However, the overlap between the 10184 synsets in the dataset from Deng et al. (2010) and the ImageNet1k dataset used to train the Inception-V1 model architecture was not complete due to the difference in time they were published. In Table 5.2 (a), the overlap between the two is shown. Of the 1000 synsets in the original ImageNet1k dataset, only 963 existed within the fall2011 version of ImageNet which were subsequently used as a starting position for our dataset. In Table 5.2 (b) the final size of the used versions of ImageNet are shown (further explained

in Section 5.1.1). Purposefully no synsets are shared between all three datasets, the ideal size of the *random* test-set was determined first to still allow the *narrow* dataset to be of the same size. The meaning of the *after-matching* column in Table 5.2 (b) is explained in the next section.

Originally, an additional experiment was designed to test the effect a larger training data-set had on the zero-shot performance leading to a total of 5579 synsets (693 original training classes, 2413 test classes (797+802+814) and an additional 2473 training classes). Although this larger training dataset was not used, the evaluation-benchmark does use all these classes in order to make the task-evaluation more difficult. See Section 5.1.3 for more details on the task evaluation setup.

(a)

	ImageNet1k	ImageNet10k	Literal ∩	Tree-overlap ∩
#Nodes	1000	10184	963	1817
In XML	1000	10155	—	—
Leafnode	650	7385	636	1393
Non-Leafnode	350	2770	327	424

(b)

	Original	After-matching
ImageNet1k	1000	693
internal	1137	797
random	1137	802
narrow	1137	814
*other	4963	3539

Table 5.2: On the left (a), one can observe the amount of leaf-nodes and internal nodes for ImageNet1k and ImageNet10k. Literal overlap indicates the amount of overlapping synsets, while the Tree-overlap also considers a synset overlapping if a more general synset is available in the ImageNet1k dataset. On the right (b), the original dataset sizes are listed before and after all processing steps as discussed in 5.1.1. The asterisk (*) indicates that this dataset was not included in the test-set directly but were included during the evaluation-setup in order to include random other classes that were not part of either the training- or test-set.

5.1.2 *Cross-modal Embedding Baseline*

An overview of how the cross-modal embedding space is created is shown in Figure 5.3. The extracted visual Inception-V1 feature representation of images from a particular synset are attempted to be matched with the language-representation of that particular synset. In ImageNet a synset contains a bag of images which represent the visual domain in our model (V in blue) while each synset also contains one or more short textual descriptions representing the class/entity name of the synset (T in yellow). These names are frequently not unique and therefore a synset-ID is used to disambiguate the classes. In addition, a more detailed textual description of the class is given. In XML these are represented by the keywords *wnid*, *words* and *gloss* respectively, a corresponding example can be seen in Figure 5.2. How this information is represented in our model can be observed in Figure 5.3. Here the vision-language pairs are projected close to each other within the language domain (yellow) by learning a parameterised function, a Multi-Layer Perceptron (MLP), to project the visual features close to their language counter-part.

```
<synset wnid="n12144987" words="dent corn,
    Zea mays indentata" gloss="corn whose
    kernels contain both hard and soft
    starch and become indented at maturity">
```

Figure 5.2: Example of an xml-entry of the ImageNet tree structure. By nesting synset definitions, the hierarchical structure of ImageNet is obtained.

Visual features were specifically extracted from the $AvgPool - 0a - 7x7$ layer of the Inception-V1 architecture resulting in a 1024-vector image representation. An overview of the Inception-V1 architecture is shown in Figure 5.4 and the model definition is available here. For a more detailed description of the model's architecture we refer to the work of Szegedy et al. (2015). This particular layer was selected due to its significant reduction

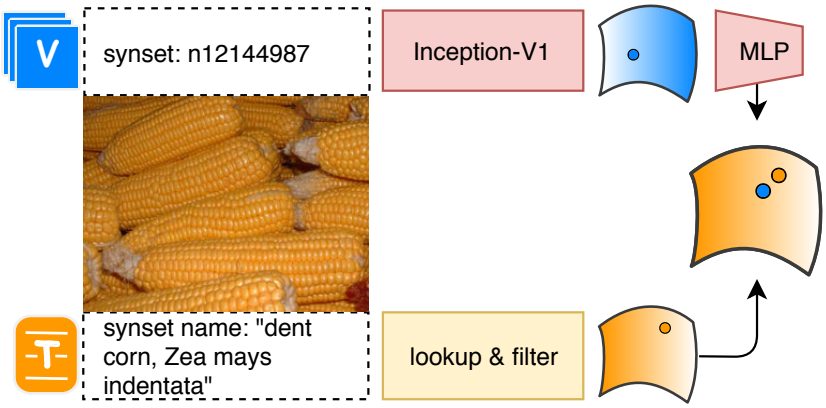


Figure 5.3: Illustration of the zero-shot evaluation setup. On the left, the images (V) and text (T) come from the ImageNet dataset where each synset consists of 200 images and the synset name describes the synset in text in possibly multiple alternative ways separated by commas. Features are extracted from the inceptionv1 network and projected down into the language manifold by a MLP. The language manifold is obtained by first matching the synset-name with the word-embedding vocabulary which requires a look-up and filtering operation. The MLP tries to minimise the corresponding projected visual and word-embedding representation of a synset.

in size compared to the previous layer which reduced the dimensionality from *7x7x1024* to *1024* through pooling and convolution operators. As a result, spatial information was traded for the benefit of being computationally more feasible to train and store. The input of the model are images that were centre cropped to $224x224x3$ after the smallest dimension of the images were first re-scaled to 256 while keeping the original aspect ratio intact. This procedure corresponds with the data-augmentation done by Szegedy et al. (2015) during the testing phase of their Inception-V1 model. The feature-representation of an image was represented by the blue dot in Figure 5.3 whereas the manifold the image got projected into represents the 1024-dimensional feature space (the manifold in blue).

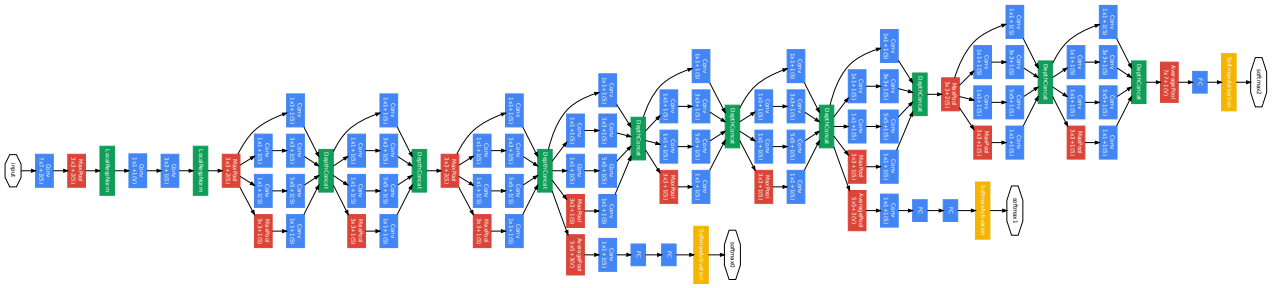


Figure 5.4: Overview of the Inception-v1 architecture with repeating inception modules. An Inception module consists of one or multiple pooling operations (red), convolutions (blue) with in the end a concatenations of the multiple parallel components (green) before connecting to the next Inception-module. For a stronger gradient there are multiple repeated softmax outputs at multiple levels in the network architecture (yellow).

The matching of the synset *wnid* or UID with a language-representation of the synset, is represented by the *look-up* operation as is shown in Figure 5.3. In ImageNet each synset UID has a *synset-name* that describes the synset in natural language text. Frequently there are multiple alternative descriptions for the synset in the synset-name which consist of one or multiple words, e.g. *toilet paper*, *toilet tissue* and *bathroom tissue* which are all alternative descriptions of the same synset (for an example of a XML ImageNet entry, see Figure 5.2). These descriptions are frequently not unique to the given synset and therefore can not be used as an UID. However, they are commonly used to represent synset-classes in image-classification

tasks as a human-readable alternative of the synset-wnid. As the overlap between the vocabulary of semantic embeddings and these textual-synset descriptions are frequently not an exact match, a lookup-operation is required that attempts to match the two. As language embedding methods use different methods in order to select their training-vocabulary, there is no one-approach-fits-all solution to match the language vocabulary with the textual synset-descriptions. For example, within our selection there is a large diversity in how these language embeddings handle capital letters, representation of spaces and frequently co-occurring n-grams within the used textual-datasets.

For the matching of the synset-name with the vocabulary of word-embeddings, two objectives were taken into consideration. First, as ImageNet is both used for fine-grained and wide-scale classification the aim was to *filter* out the fine-grained classes and only keep one synset representing the whole group. Arguably this is a subjective step as it is dependent upon how semantically similar two synsets are perceived. Therefore this step was carried out manually. For example, when the synset-description "Chihuahua" was found in the ImageNet10k dataset as well as the synset-description "dog", only the latter was kept. ImageNet1k contains many synsets centred around some specific species in order to carry out fine-grained image classification. This can be roughly observed by the amount of non-leafnodes included in the dataset ($\approx 350/1000$) in Table 5.2(a). For our task, fine-grained classification performance was not assumed to be important due to (1) the unconstrained nature of language in the TALL-task which stresses *general* video understanding. Also, (2) as the TALL-task already focuses on a single video to search a particular event in, this limits the probability of a similar event happening in the same video which would require more fine-grained visual representations.

The second objective was to ensure that the comparison between the word-embeddings was fair. For this, the same lookup operation was applied to all language-embeddings independent of whether the synset-word description could be directly matched with only *some* of the vocabulary of the different word-embedding methods. This was deemed necessary as we expected that using different lookup criteria such as the averaging of individual words if not the entire textual description could be matched directly, would greatly impact the zero-shot evaluation score of this particular class. This could create an unfair advantage to the language embeddings that actually contained the synset-textual description directly compared to the ones who did not. The precise methodology to process the *words* of synsets in Figure 5.2 is explained now.

To go from the *words* attribute to words that can be matched in the word-embeddings' vocabulary a multitude of steps were

taken. First, the word-embedding vocabulary was all lower-cased. If duplicates were introduced, only the original word-vocabulary was kept. The same procedure was adhered to in all subsequent steps when duplicates were introduced. Spaces were replaced to underscores and apostrophes were removed. It was found that the ImageNet synset *word*-descriptions contained spaces in about 35% of all cases, see Figure 5.5. From a closer manual inspection, this was found to mostly occur because animal names generally tend to consist of a compound of multiple words originating from a Latin-origin. The *words*-attribute was split on commas to separate the alternative synset descriptions, after which the same processing criteria were carried out as for the word-embedding vocabulary. The multiple word-representations of the synsets were then sorted on the number of spaces and character length in ascending order. It was expected that the shortest synset-description was most frequently used in written-text leading to more accurate word-embeddings and more frequent appearances in the DSM vocabulary.

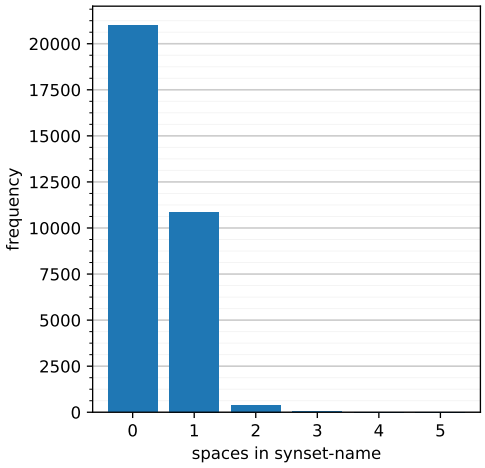


Figure 5.5: The amount of synsets containing n number of spaces in the whole ImageNet (32297 classes). Used as illustration for difficulty of matching word-embedding vocabulary with ImageNet synset class names.

After this step, for all synsets each word description was looked up in all the different language embeddings methods; *Glove*, *Word2Vec*, *Lexvec*, *Numberbatch* and *ours* of which the last two are dependent on the node names of ConceptNet (discussed in Section 5.2). Prior to this step, as aforementioned, all the synsets that were centred around fine-grained image-classification were manually mapped to their parent species, after which duplicates were removed to not unbalance the number of images (200) per synset. Consequently, the appropriate word-embedding representation for each synset was attempted to be found.

For each synset the processed *words* were matched with all word-embeddings at once, if a match with the vocabulary of all embeddings was available this representation was used. On average there are 1.85 different *words* descriptions per synset. If no direct match was found for all embeddings at once, an additional effort was made to *redefine* the descriptions manually to a suitable class-name one-word synonym that was available in all embeddings. As this is a time-intensive task, this was only done for the examples in the training-set to ensure that the learned cross-modal embedding space was accurate. If no such alternative could be found, instead the descriptions with multiple words were parsed to the word-level after which the synset-representation was defined as the average of the individual word-embedding representations. For instance *toilet paper* would become $\frac{1}{2}Emb(toilet) + \frac{1}{2}Emb(paper)$ as a representation. In the rare cases that this still lead to words not present in the vocabulary, the maximum amount of matches overall word-embedding vocabulary was chosen, and individual Out of Vocabulary (OOV) words were discarded (e.g. only

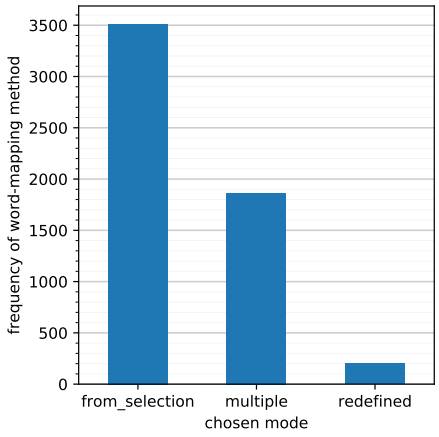


Figure 5.6: Matching operations used to match the ImageNet synset names and vocabulary of Word2Vec, LexVec, Numberbatch and Glove. *from_selection* : was used if one of the synset-names was present in all of the word-embedding vocabularies. *multiple* : if all the individual words of one synset-name were present in all language-embedding vocabularies, the synset representation was obtained by averaging the individual embeddings. *redefined* : if a 1-word synonym was found for the synset-descriptions or the class was mapped to a more general parent class. Statistics based on all 5579 synsets.

the embedding $Emb(toilet)$ was used if $Emb(paper)$ was not found in the DSM vocabulary). In Figure 5.6 one can observe how many times each operation was applied.

5.1.3 *Training Objective & Evaluation Benchmark*

To evaluate to which extent the parameterised *MLP*-projection in Figure 5.3 was able to transfer knowledge from the corresponding visual-language pairs to unseen pairs in a zero-shot setting, first the cosine-distance between the projected image and each synset language representation was calculated. Thereafter, the cosine-distances were sorted and ranked according to their similarity in ascending order. The rank at which the correct image-synset appears was used as a measurement for success (lower is better) and was captured in terms of mean Average Rank (mAR), and the mAP@10. For both cases, the full dataset of 5579 synsets was used that included the training-synsets, three test-sets and 2413 unrelated other synsets as discussed in Section 5.1.1.

Now follows a more detailed explanation of how the mean average precision was calculated. For this we first introduce the notation as displayed in Figure 5.7. Starting with a particular image I_i and the function $F(I)$ which represents the Inception-V1 feature extraction at the *AvgPool_0a_7x7* layer, a 1024 image feature-vector v is obtained. Thereafter the parameterised function P_θ by θ takes in v and learns to minimize the distance $P_\theta(v)$ and the corresponding word-embedding feature representation of a synset s. This word-embedding feature-representation is obtained by the looked operation $L(s)$ as explained in the previous section. The result of the previous section is the dataset D that consists of corresponding image and synset pairs indicated by the same index i:

$$D = \{I_i, S_i\} \tag{5.1}$$

The objective for the MLP network is to minimise the loss-function $\mathcal{L}$ below. Here, either the cosine or contrastive loss-function was used represented by either $\gamma = 0$ or $\gamma > 0$ respectively:

$$\mathcal{L}(\theta) = -\sum_{(i,s)\in D} cos(i, s; \theta) + \gamma \mathop{\mathbb{E}}_{s_{neg}\sim U(S\setminus\{s\})} [cos(i, s_{neg}; \theta)] \tag{5.2}$$

The cosine similarity function is defined as:

$$cos(I, S; \theta) = \frac{P_\theta(F(I))^T \cdot l(S)}{||P_\theta(F(I))|| ||l(S)||} \tag{5.3}$$

Where P_θ is the projection function that projects the image-feature representation $F(I_i)$ to the image domain and $l(S_i)$ is the lookup function that returns the language-feature representation of the synset s. The loss is then minimised by the

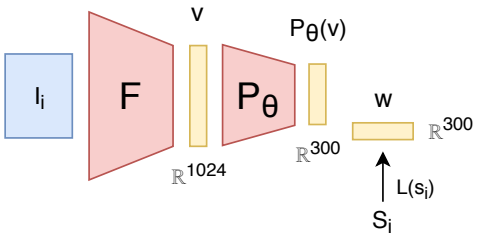


Figure 5.7: Schematic overview of the MLP-projection network as was first visualised in Figure 5.3. This figure is best understood in conjunction with the notation introduced in text on the left-hand side.

following equation.

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}(\theta) \tag{5.4}$$

With this trained MLP network finally the performance in terms of zero-shot performance can be calculated. A ranking function is introduced that takes in images and synsets and returns the rank at which the corresponding image-synsets are found on average.

$$rank : I \times S \rightarrow \mathbb{N} \tag{5.5}$$

The rank is calculated as follows. First the cosine similarity between the projected image $P_\theta(F(I_i))$ and all synset S word-embedding feature representations is calculated $l(S)$. Thereafter the rank is assigned to be equal to the amount of non-corresponding I_i and language synset-representations $l(s_j)$, where $j \neq i$, ranked higher than the corresponding one where $j = i$.

$$\begin{aligned} rank(I_i, S) = |s' \in S; cos(P_\theta(F(I_j)), l(s'); \theta) \\ < cos(P_\theta(F(I_i)), l(s_i), \hat{\theta})| \end{aligned} \tag{5.6}$$

However, this only calculates the rank of a single image i. The total score is calculated by averaging over all images within one synset, after which these scores are averaged over all synsets of the particular test-set of interest to obtain the mAP,

$$mAP = \frac{1}{|S|} \sum_{s \in S} \frac{1}{|I_s|} \sum_{i \in I_s} rank(i, S) \tag{5.7}$$

Where I_s is defined by the following equation.

$$I_s = \{i | (i, s) \in D\} \tag{5.8}$$

The mAP@10 was reported instead of @5 or @1 as the other evaluation cut-offs showed significantly higher variance presumably due to the high amount of training and test-classes (5579). Random performance would given our evaluation task only have a mAP@10 of 0.179% (10/5579*100).

5.1.4 *Architecture Selection & Training*

As the objective of Experiment I is to obtain a zero-shot performance score to compare the ability of different language embeddings to be paired with their visual correspondences, an appropriate MLP model architecture should be chosen. First, different MLP network depths and layer sizes were tested for improving the zero-shot evaluation scores. A two-layer MLP of size $300x300$ performed best on the evaluation benchmark suite which was explained in more detail in Section 5.1.3. More layers improved training performance but decreased the generalisability to unseen classes indicated by a lower zero-shot

performance score. This architecture was used as a starting point to compare the two different loss-functions, the cosine similarity and contrastive loss-function, under a variety of parameter settings. These loss-functions are now described in more detail than was initially shown in Equation 5.2.

Contrastive Loss A contrastive loss-function tries to force corresponding pairs to be semantically close while artificially enforcing non-corresponding pairs to be as dissimilar as possible. Given an input pair I_i, s_i, the loss-function becomes the following.

$$loss(I_i, s_i) = \begin{cases} (1 - cos(I_i, s_i)) \cdot w_p & \text{if } i = j \\ \max(0, (\cos(I_i, s_i) - m) \cdot w_n) & \text{if } i \neq j \end{cases} \tag{5.9}$$

where m is the margin which decides how much error is allowed for positive examples. In literature, this is frequently set around 0.2. Empirically it was found in our experimentation that positive examples were found to be significantly more important for zero-shot evaluation accuracy. As a result, an extra component was added that weighted positive w_p examples differently from negative ones w_n. For our use-case w_p was fixed to 1 and w_n was considered a free parameter.

Cosine Distance Loss The cosine distance is a measurement to calculate the similarity between two vectors in a multi-dimensional space. It is dependent upon the cosine similarity which calculates the similarity between two vectors $\vec{a}$ and $\vec{b}$ as follows,

$$similarity(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}||||\vec{b}||} \tag{5.10}$$

With the cosine distance being defined as:

$$distance(\vec{a}, \vec{b}) = 1 - similarity(\vec{a}, \vec{b}) \tag{5.11}$$

In Figure 5.8 one can observe the difference between the Euclidean distance and cosine similarity. For the cosine similarity what matters is the angle between the two vectors, due to the normalisation factor $||\vec{a}||||\vec{b}||$ the length of the individual vectors does not affect the cosine similarity. The cosine similarity is defined between -1 and one while the cosine distance is defined between 0 and 2.

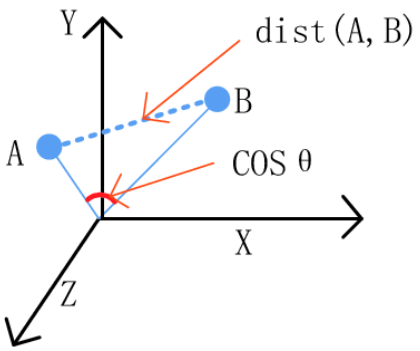
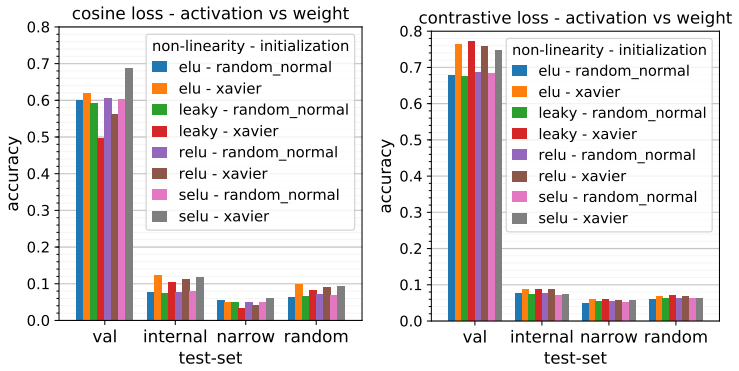


Figure 5.8: Difference between cosine and euclidean distance.

Parameter-selection For all parameter settings 20 epochs were run, starting with a learning rate of 0.001 with for the contrastive loss-function *w_neg* and *w_pos* set to 1. Each synset

contained 200 images. For training and testing 180 images per synset were used, 15 for the final testing and 5 for validating the model performance during fine-tuning. As testing the zero-shot performance is a relatively expensive operation as it first requires training the MLP model after which the evaluation benchmark has to be run for all language embeddings, the decision was made to do this on a relatively small dataset while still ensuring that the results were consistent over multiple runs. As the zero-shot performance was assumed to be highly dependent upon the selection of synsets, the decision was made to use the same synsets both for fine-tuning and final testing. Potential problems of this approach are discussed in Section 7. The order in which parameters were tuned were: activation and weight initialisation, margin, weight negative and positive examples, learning rate, dropout rate and batch-normalisation. Each experiment leads to one parameter being changed. The parameters were fine-tuned on the Glove embeddings specifically. This could potentially lead to over-fitting towards this specific language embedding. However, the obtained results on the validation and test-set showed similar relations which partially neglected this concern (Results 6).



(a) cosine: activation-function vs. weight initialization

(b) contrastive: activation-function vs. weight initialization

Figure 5.9: Parameter selection: activation vs. weight initialization method. Results are listed on the validation-set and the 3 different test-sets.

First the non-linear activation-functions; elu, leaky, relu and selu were tested under different weight initialization methods; random normal or xavier. The results are shown in Figure 5.9(a) and (b). Selu with xavier initialization performed best on the narrow test-set with the cosine loss-function with minimum differences to the contrastive loss-function, therefore this combination was selected for both. Thereafter, the parameters m and w_n were fine-tuned as displayed in Figure 5.10 and 5.11(b) respectively. Based on these findings, the following settings were used; $m = 0.15$ and $w\_neg = 0.0064$.

Thereafter the learning rate was fine-tuned as shown in Figure 5.11(a) and (b). Batch-normalisation was applied before or after each layer while temporally disabling dropout. However, the inclusion of batch normalization greatly reduced the performance on the test-sets (5.11(a)). One possible reason for this

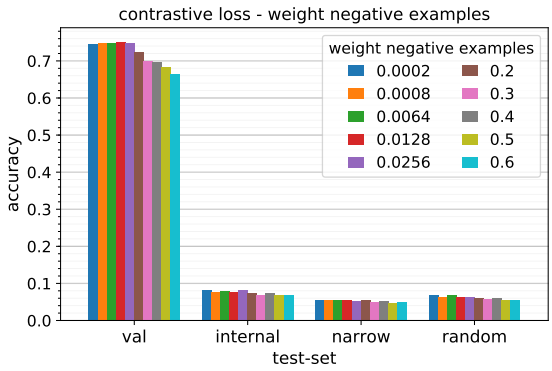
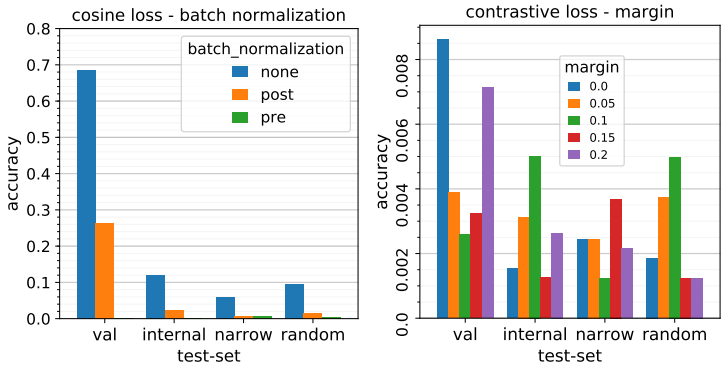


Figure 5.10: Parameter selection: weight of negative and positive examples. Results are listed on the validation-set and the 3 different test-sets.

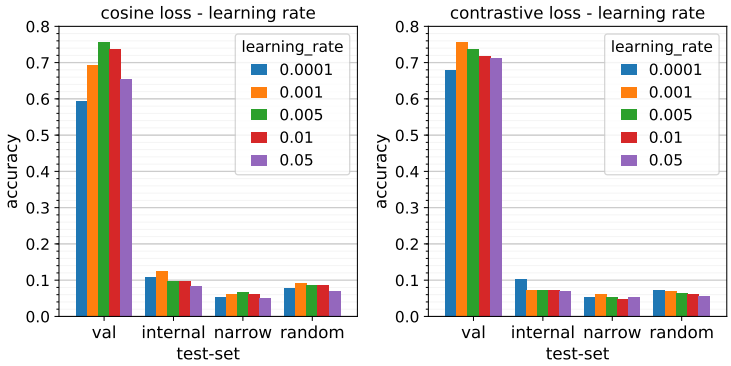


(a) contrastive: margin

(b) Effect of pre- and post-activation batch-normalization in the *MLP* module.

Figure 5.11: Parameter selection: batch normalization (left) and margin of the contrastive loss-function. Results are listed on the validation-set and the 3 different test-sets.

is that on the test-sets the feature-distribution is significantly different from the training-set due to the classes being disjoint with the purpose of zero-shot evaluation. However, this can not explain why the application of batch-normalisation performed significantly worse on the validation-set. Based on this finding batch-normalisation was not used in any of the further experiments. A learning rate of 0.005 was found to work best for cosine similarity, and 0.001 for the contrastive loss-function.

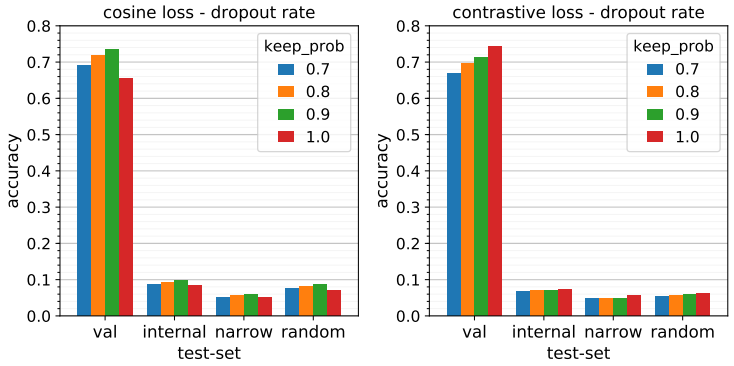


(a) cosine: learning rate

(b) contrastive: learning rate

Figure 5.12: Parameter selection: learning rate. Results are listed on the validation-set and 3 different test-sets.

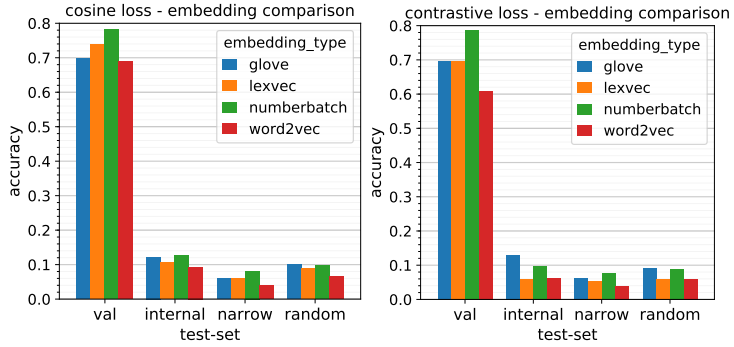
With all the settings applied, the results for all the word-embeddings are shown in Figure 5.12 on the validation-set with five images per synset. In Table 5.15 the final settings are shown.



(a) cosine: dropout rate

(b) contrastive: dropout rate

Figure 5.13: Parameter selection: dropout. Results are listed on the validation-set and 3 different test-sets.



(a) cosine: with final settings

(b) contrastive: with final settings

Figure 5.14: Final mAP on the validation zero-shot evaluation dataset for both the cosine and contrastive loss-function.

In Figure 5.15 one can observe the effects that the averaging of word-embeddings have on zero-shot and validation-set performance. It is important to note that all redefined nodes come from the validation set of the training-set and therefore perform significantly better. For the zero-shot performance, one can observe that averaging over multiple words to obtain a synset word-representation is not recommended with most of the distribution's weight centred around the lower end of the spectrum (in green).

Parameter	Setting-cosine	Setting-contrastive
epochs	20	20
hidden-layer-1	300	300
hidden-layer-2	300	300
keep-prob	1	1
batch-normalization	None	None
weight-initialization	xavier	xavier
learning-rate	0.001	0.005
activation-function	selu	selu
margin	N/A	0.15
weight-negative examples	N/A	0.0064
weight-positive examples	N/A	1
batch-size	1024	1024

Table 5.3: Final settings of the 2-layer MLP.

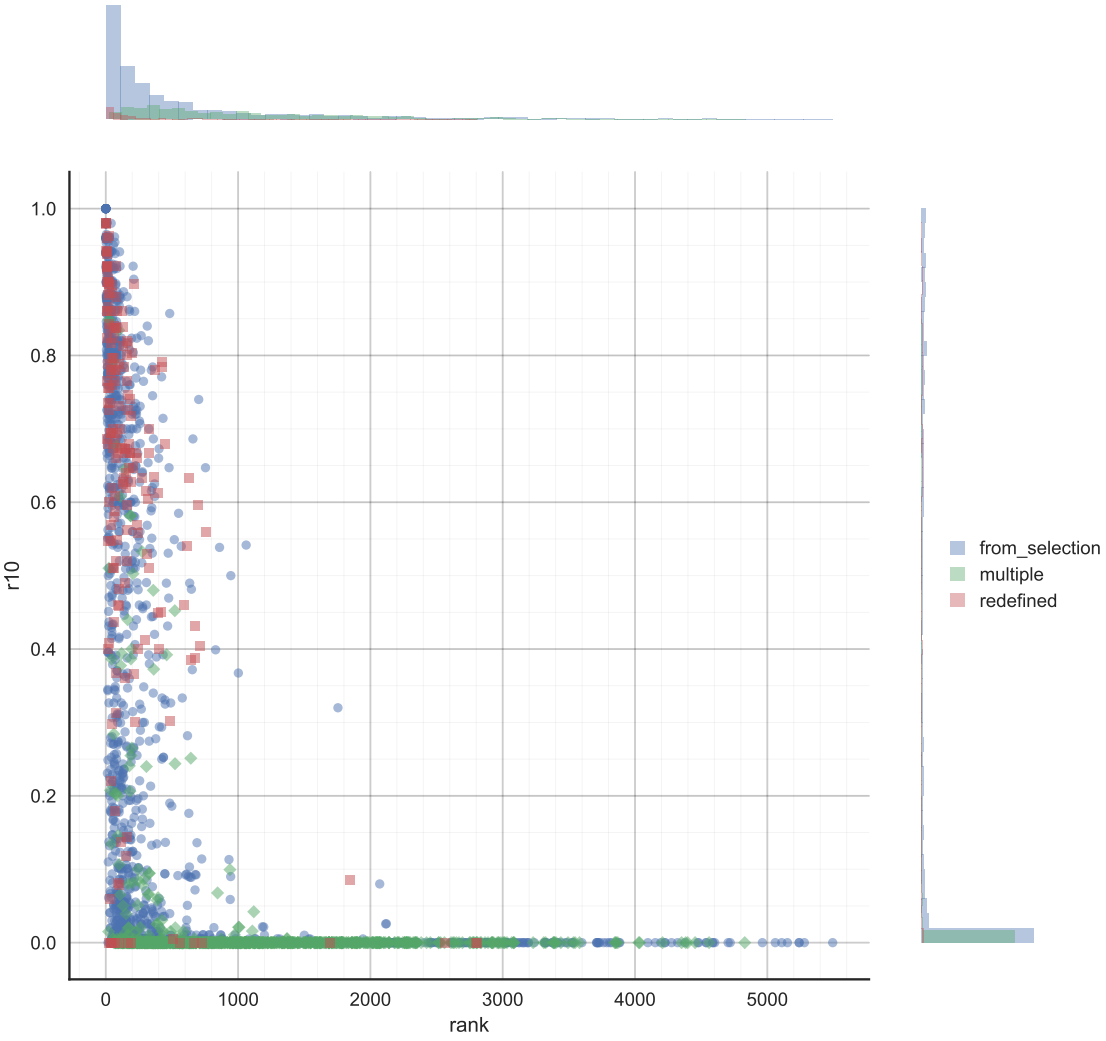
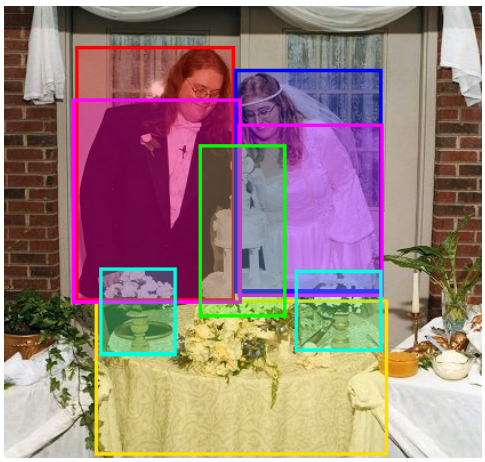


Figure 5.15: The effect of the missing word replacement from Figure 5.6 on rank@10 and average rank.

5.1.5 *Qualitative Analysis using Flickr30k*

While the zero-shot performance scores obtained in Experiment I gives a general sense to the extent the structure of the language embeddings can be be matched with visual features, it provides little insight into which relations are easily accessible. Therefore an additional experiment was conducted using the Flickr30k dataset with the objective to gain more quantitative information of the different relationships that were easily accessible in the cross-modal embedding space that was obtained in the Experiment I. For this the sentences were POS-tagged in order to obtain more insight into which relationships are accessible in the obtained cross-modal embedding space for the different language embedding methods. Plummer et al. (2015) introduce the Flickr30k dataset to provide a benchmark for sentence-based image descriptions by providing images with five different sentence descriptions that describe in text what visually happens within the image (Figure 5.16). An additional benefit that is obtained when using this particular dataset is the increased vocabulary size and variety of images. The entire vocabulary size is 11807 which is roughly a factor 10 higher than the TACoS and Charades-STA datasets used to evaluate the TALL-task.

To calculate the sentence similarity for each image, the pre-trained cross-modal embedding space that was obtained in Experiment I was used. The sentences of the Flickr30k dataset were POS-tagged and parsed to the word-level. Thereafter, for each word the cosine similarity was calculated with a random but fixed selection of 1000 images using an average score of the five image descriptions per image. With the POS-tags more specific information was expected to be obtained about the ability of the MLP to extract meaningful relationships between the different word-types and the image. For example it was expected that nouns-image pairs contained the strongest similarity of the different POS-tags, while verb-image pairs had stronger similarity in language embeddings obtained using relational knowledge due to their central position within ConceptNet. The frequency of the POS-tags in our dataset-selection can be seen in Figure 5.17. The sentence similarity was obtained by taking a word-image cosine similarity average and the image-sentence similarity scores were ranked for all 1000 images and sentence combinations. Ideally the highest similarity was given to the corresponding image-sentence pairs. However, the model's ability to rank corresponding sentence-image pairs higher than non-corresponding pairs was insufficient (Section 6). This indicates that the cross-modal embedding space obtained in Experiment I was not expressive enough to be used on real-sentences to allow for any further qualitative analysis. Therefore this experiment was only included to demonstrate the idea of using the Flickr30k dataset for potential further



A couple in **their wedding attire** stand behind **a table** with **a wedding cake** and **flowers**.
A bride and **groom** are standing in front of **their wedding cake** at **their reception**.
A bride and **groom** smile as **they** view **their wedding cake** at **a reception**.
A couple stands behind **their wedding cake**.
Man and **woman** cutting **wedding cake**.

Figure 5.16: Each image in the Flickr30k dataset is described by 5 different people resulting in a diverse sentence-annotated dataset.

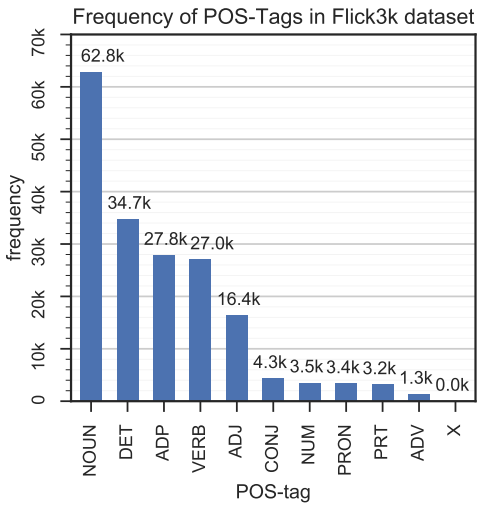


Figure 5.17: POS-tag frequency in the used Flickr30k subset.

qualitative analysis.

5.2 II - GraphSAGE-Conceptnet Embeddings

In Experiment I a cross-modal embedding space was obtained to indicate the ability of the language embeddings to be used in a zero-shot performance task-setting. In this experiment, we attempt to create our own language embeddings that hypothetically are more structured and visually centred to be better suited for the TALL-task.

This experiment can be broken up in three distinctive parts. First, ConceptNet is analysed in Section 5.2.1 and compared to other datasets that were used by GraphSAGE. Thereafter in Section 5.2.2 a selection of ConceptNet was made and the GraphSAGE dataset was created. For this, a similar problem had to be resolved as in Experiment I in which now the vocabulary of ConceptNet had to be matched with one of the DSM methods such that node-embedding features could be added. Lastly, we discuss how we trained the GraphSAGE algorithm and performed the parameter search in Section 5.2.3.

5.2.1 ConceptNet Analysis & Graph Comparisons

Hamilton et al. (2017) applied their newly introduced GraphSAGE algorithm on a variety of different tasks, including citation and protein-protein interaction predictions. This raises concerns about whether this approach is also suitable to be applied to a different domain (ConceptNet) and with a different objective (to obtain language embeddings). To address the former issue, first the *conceptnet-assertions-5.6.0.csv* version of ConceptNet (available here) was analysed in terms of overall size and available relationship types after which this was compared to other datasets. The used version of ConceptNet contains 32755210 rows with concepts in 78 different languages with each entry representing a tertiary relation $< subject, relation, object >$. Two example rows of the raw dataset are shown in Figure 5.4.

0	1	2	3	4
/a/[/r/RelatedTo/, /c/fr/rÃl'empaffera/v/, /c/fr/rÃl'empaffer/]	/r/RelatedTo	/c/fr/rÃl'empaffera/v	/c/fr/rÃl'empaffer	{"dataset": "/d/wiktionary/fr", "license": "cc:by-sa/4.0", "sources": [{"contributor": "/s/resource/wiktionary/fr", "process": "/s/process/wikiparsec/1"}], "weight": 1.0}
/a/[/r/EtymologicallyDerivedFrom/, /c/io/soneto/, /c/es/soneto/]	/r/EtymologicallyDerivedFrom	/c/io/soneto	/c/es/soneto	{"dataset": "/d/wiktionary/en", "license": "cc:by-sa/4.0", "sources": [{"contributor": "/s/resource/wiktionary/en", "process": "/s/process/wikiparsec/1"}], "weight": 1.0}

Table 5.4: Example of two raw dataset-entries of the *conceptnet-assertions-5.6.0.csv* version of ConceptNet.

For two reasons the decision was made to focus only on the English vocabulary. First, this dataset size is orders of magnitudes more substantial than the largest dataset GraphSAGE was applied on which were already considered large, see Table

5.5. Second, in our approach we add language embeddings to the nodes in ConceptNet in order to combat the relationship-sparsity in ConceptNet, which is a common difficulty in KBs. By focusing only on the English vocabulary, this avoids the difficulty of finding accurate word-embedding representations for non-English languages with a subsequent challenge being how to relate the word-representations of different DSM methods in one joint representation.

As ConceptNet contains relationships between concepts from different languages, considering only concepts that go *from* and *to* the same language resulted in only 4209727 rows and 1856150 unique concepts. This selection was used as a starting point and was later reduced to 1029619 by additional filtering steps which will be discussed shortly.

Dataset ↓ Statistic →	Nodes	Avg Degree	Features Source	Dimension
Citation	302424	9.15	word2vec	300
Reddit	232965	492	GloVe CommonCrawl	300
PPI	2373	28.8	variety of molecular info	121
GraphConceptNet (ours)	1029619	5.11	NumberBatch 17.06	300

Table 5.5: Comparison of the datasets used by Hamilton et al. (2017) with our selection of ConceptNet.

relation	start	end	dataset	weight	start_word	start_pos	end_word	end_pos
synonym	/c/en/nathan/n	/c/yi/ŒăŒłŒ§	/d/wiktionary/en	1.0	nathan	n	ŒăŒłŒ§	
is_a	/c/en/false_sago/n	/c/en/sago_palm/n	/d/wordnet/3.1	2.0	false_sago	n	sago_palm	n
related_to	/c/en/here	/c/en/place	/d/verbosity	1.179	here		place	
related_to	/c/en/claudin/n	/c/en/protein	/d/wiktionary/en	1.0	claudin	n	protein	
synonym	/c/en/magnolia/n	/c/fi/magnolia	/d/wiktionary/en	1.0	magnolia	n	magnolia	

Table 5.6: Example of five cleaned entries of the *conceptnet-assertions-5.6.0.csv* version of ConceptNet.

The original *csv-file* was processed of which a sample is shown in Table 5.6. The *start*, *end* and *relation* attributes together represent one tertiary relation $< concept\_form, relation, concept\_to >$ in the dataset. The *weight* variable is unused here (usually 1.0), but reflects a reliability score of the source this row-entry originated from. The *start_word* is the actual concept-word separated from the *start* attribute which also has the POS-tag if it was available in *start_pos*. As current DSM methods do not disambiguate between different word-usages (*homographs*), this attribute was not used. With the python library Networkx introduced by Hagberg et al. (2008) the tertiary relationships were added back to a graph-structure which significantly increased the speed in which graph-information could be obtained.

In Table 5.5 one can observe the difference in size (*Nodes*) and connectivity (*Avg Degree*) together with the node-feature representation size (*Features Source*) of the datasets used in the original paper from Hamilton et al. (2017). Here one can observe that ConceptNet is considerably larger than the other datasets with significantly lower connectivity while being on par with the node feature-dimensionality size. A more in-depth analysis of the distribution of the average node degree showed that the distribution is heavily tailored towards nodes having

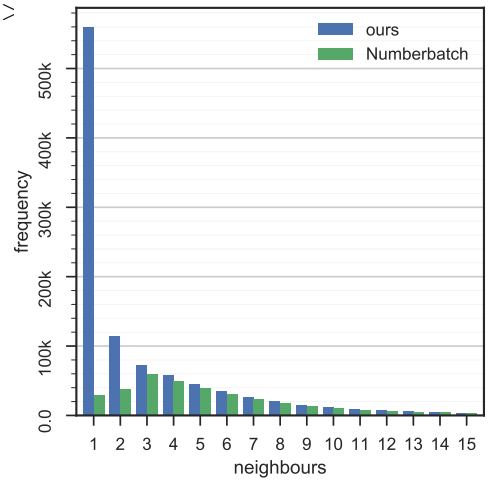


Figure 5.18: Distribution of neighbours in our selection of ConceptNet compared with that of Numberbatch. From Numberbatch only the overlapping vocabulary is shown.

only one neighbour (Figure 5.18). As Numberbatch relies on the ConceptNet hierarchy and only selects specific concepts thereof, one question that could be raised was whether the distribution of neighbouring nodes was the same. In Figure 5.18 one can observe that in Numberbatch a significant portion of nodes that contained only 1 or 2 neighbours was left out while containing a similar absolute amount of nodes for all other neighbours.

How dense the information was clustered around only a select few number of nodes was also deemed important for obtaining semantically meaningful relations using our approach. To visualise how the 4209727 tertiary relations were distributed around the 1856150 unique nodes, the percentage of relations covered by nodes that contained up to n-neighbouring nodes was shown in Figure 5.19. Here, one can observe that the cumulative sum of nodes with up to 30 neighbours reaches 98.19% of all nodes while the total amount of edges covered by these nodes is only 59.52% of the total amount of rows. This means that a significant portion of information is centred around only very few concepts. As words that are more frequently used in practice tend to be better represented in knowledge graphs as well as DSM approaches and intrinsic evaluation benchmarks, it was expected that this would yield more qualitative embeddings and higher performance.

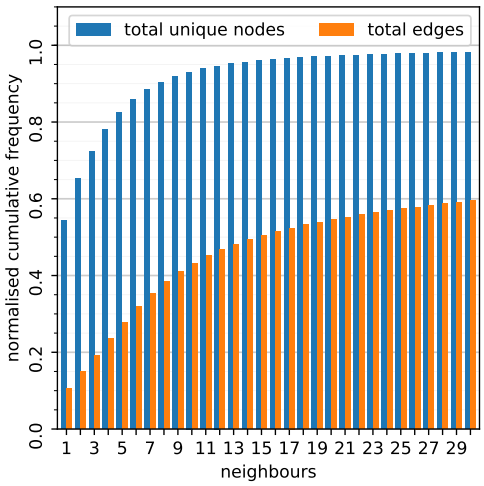


Figure 5.19: The information that is contained in our selection of ConceptNet as a function of the nodes that contain at most n neighbours. In orange one can observe the % of the dataset that is covered by the nodes.

5.2.2 *OOV Matching & Dataset Creation*

For the creation of the GraphSAGE dataset, the aforementioned English-only selection of ConceptNet was further refined. First, it was assumed that the 1856150 unique nodes could not all be reached by traversing through all the edges. To remove sub-graphs that were potentially irrelevant for our specific sub-domain of event-localisation, only nodes were included that were reachable starting from the 5579 synsets within our ImageNet zero-shot dataset-set. For this, the directionality of relationships was not taken into account, which is further discussed in the next paragraph. This selection of synsets was deemed a decent starting point as ConceptNet contains 99.92% of our selection of synsets which in addition also have a high 24.78 relationships per node. As these were only centred around objects, it was expected that these had strong visual correspondences useful for event-localisation in videos.

Subsequently, as GraphSAGE as presently constructed does not differentiate between (1) different relationship types or (2) edge directionality, decisions had to be made about whether to include non-symmetric relationships and how to handle the different edge types as observed in Figure 4.2. For example the tertiary relation <*pan, used_for, cooking*> is different from <*cooking, used_for, pan*> as the relationship is not symmetric. Aggregating node neighbourhood information from asymmetric

relationships, e.g. *is_a*, as if they were undirectional could be problematic. In specific, to not take directionality into account means that the node *dog* which is two hops away from a giraffe is considered equally relevant for the node-embedding feature representation as a more specific species of dogs that is also two hops away. Intuitively the former should be taken into account less than the latter for the feature representation of *dog*. Therefore, the decision to which relationships and concepts to include given these modelling constraints is assumed to have a significant effect on the obtained language-embeddings.

In Figure 5.20 one can observe how many edge traverses are required to reach n unique nodes in the case in which edges are assumed to be directed or undirected starting from the 5579 concepts in our zero-shot dataset. When the edge-direction is taken into account, a neighbouring node can not be reached if it is against the direction of the relation. Here one can observe that the number of unique nodes that are reachable decreases significantly when edge directionality is taken into account. The same pattern was observed for the amount of included edges. In Figure 4.2 one could have observed that a significant part of the relationships in ConceptNet are asymmetric. Based on these observations the decision was made to apply no further filtering step to preserve the vocabulary size and more importantly the node degree. This resulted in the final selection of ConceptNet consisting of 1029619 unique nodes and 3098816 rows.

Another important question is how this subset of concepts in ConceptNet can be matched with the language vocabulary of DSM models to enrich the node representation in GraphSAGE with word-embeddings. The sparsity of the amount of information available for most concepts as was revealed in Figures 5.18 and 5.19, arguably requires the significantly more dense feature representation created using distributional approaches. One reason for the large concept- and word-embedding vocabulary mismatch is that ConceptNet allows concepts to consist of multiple words separated by underscores, whereas in the word-embeddings we used this was not the case for all except *Word2Vec* and *Numberbatch*.

In Figure 5.21 an overview is provided of the overlap of our ConceptNet vocabulary and the vocabulary of the different word-embedding methods either with undirected edges or directed edges. Here one can observe that Numberbatch has a significantly larger vocabulary overlap than any of the others language embedding vocabulary with our subset of concepts in ConceptNet. Therefore the language embeddings from Numberbatch were selected to be used as the node-feature representations of ConceptNet. In addition, Numberbatch outperformed many of the DSM in the intrinsic word-evaluation methods (Table 6.2), indicating that the embeddings are more

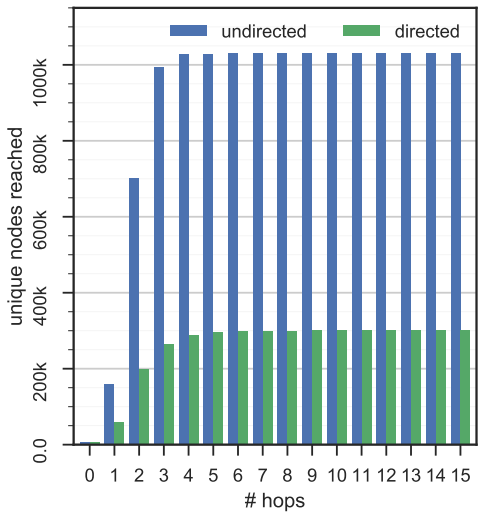


Figure 5.20: Amount of unique nodes reached starting from with the 5579 synset in our zero-shot dataset and traversing n-hops through the edges in ConceptNet. In the *undirected* case, all edges can be traversed over whereas in the *directed* case the directionality of the edges are taken into account.

in line with how humans judge similarities between words. The structure of ConceptNet was already used in the process of creating the Numberbatch word-embeddings, therefore our approach uses ConceptNet twice. First indirectly with the useage of Numberbatch which used already used the structure of ConceptNet in its creation. Secondly, in our approach where we use the structure of ConceptNet in GraphSAGE to aggregate local-neighbourhood node information. To guarantee that our obtained embeddings do not blatantly copy the Numberbatch representations to obtain the observed results (6), this is further analysed and discussed in the Discussion Section (7).

As seen in Figure 5.21 still a significant part of the vocabulary of ConceptNet that was used here (*undir*) could not be matched with the semantic word-embedding vocabulary. Despite 91.66% of the Numberbatch vocabulary being available in our ConceptNet selection, this overlap was only 37.14% of the total vocabulary size of our significantly larger ConceptNet concept selection. Therefore, three alternative methods were used to replace the 63.86% missing Concept node feature-representations; zero vectors, node-embedding averages or local neighbourhood averages. The algorithm for the latter can be found in Algorithm 2. Speer and Lowry-Duda (2017) faced a similar problem of matching the vocabularies from different DSM approaches and used a more intelligent method to achieve this. Due to time-constraints this was left for future work.

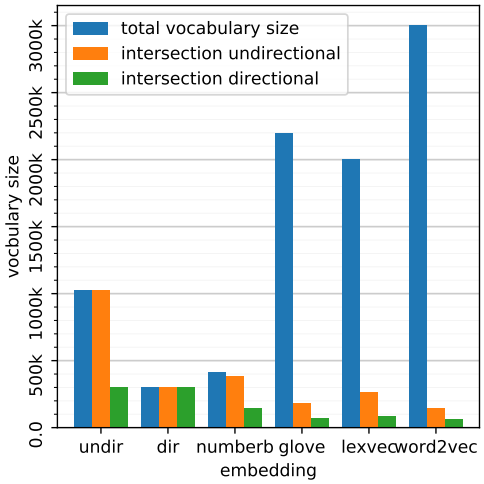


Figure 5.21: Overlap of the vocabulary for our ConceptNet selection using the *undir*ected and *dir*ected dataset as shown in Figure 5.20. Significant vocabulary overlap occurs between the Numberbatch vocabulary and our ConceptNet concept selection. In the end the *undir* ConceptNet concept selection is used.

Algorithm 2 Missing word-embeddings by neighbourhood average algorithm, called: *neighbourhood_average*)

Input : A, adjacency matrix of neighbouring nodes;
s, binary vector indicating whether word in A is in Numberbatch or has obtained a representation by neighbours.
word_vecs, all ConceptNet word-embeddings with zero vectors for unknown embeddings in Numberbatch

Output : Word-embedding for each word in ConceptNet

for i ... Iterations **do**
  $s = A.dot(s)$
  $word\_vecs = A.dot(word\_vecs)$
  $word\_vecs = divide(word\_vecs.T, s, where(s! = 0)).T$
  $s = clip(s, max = 1)$
  $word\_vecs[ids\_in\_numberbatch] = numberbatch\_vecs$

The aforementioned ConceptNet node selection and node-feature representations lead to the dataset required for the GraphSAGE model input. The required files are shown in Table 5.7. The ConceptNet sub-graph with 1029619 nodes and 3098816 rows was represented in the *G.json* file. As for our

purpose a closed world assumption can be made, where the generalisation to unseen nodes is not required, the decision was made to not use the *val* or *test* attribute as this leads to optimal performance[1]. Instead, the objective here was to maximise the word-embedding quality concerning intrinsic evaluation benchmark scores. The *id_map.json* specifies how the node-ids in the *G.json* are mapped to consecutive numbers. This is important as the *feats.npy* uses this *id_map* to find the index of the corresponding distributional semantics word representation. The *class_map* can be used to map back these ids to corresponding classes. For our case the *id_map* and *class_map* are an identity map as the *G.json* was used to reconstruct the node-id to the corresponding concept-name.

[1] Hamilton et al. (2017)

Lastly, the *walks.txt* is a file that contains random walks for each node that is consequently used to sample neighbouring nodes from for the particular node of interest. This is used as a sampling technique of a node's local neighbourhood, with negative samples being represented by nodes not in the local neighbourhood. Hamilton et al. (2017) provide a script for obtaining these random-walks and negative examples after which these are fixed for training. Acquiring positive and negative samples before training time lets the algorithm run about $100 - 500x$ faster in practice[2].

[2] Hamilton et al. (2017)

file	purpose
G.json	networkx-specified json file describing the input graph, each node has a 'val' or 'test' attribute.
id_map.json	dictionary mapping the graph node ids to consecutive integers
class_map	dictionary mapping the graph node ids to classes
feats.npy	node features with indices corresponding with range the id_map maps to
walks.txt	a text file specifying random walk co-occurrences

Table 5.7: GraphSAGE model input specifications.

5.2.3 *GraphSAGE Training & Parameter Selection*

Hamilton et al. (2017) provided an unsupervised and supervised version of their algorithm. Here, the unsupervised version of GraphSAGE was used that uses an unsupervised loss-function that specifies that the local neighbourhood of a particular node should have feature-representations that are more alike than distance regions. To obtain the best possible representation of nodes for TALL-task, first the parameters were tuned on the 17 different intrinsic evaluation metrics. The benchmark test is available here. It was assumed that embeddings that performed better on these intrinsic evaluation benchmarks would perform better to the downstream performance task. The main benefit of this approach is that testing on intrinsic evaluation metrics is faster as they do not require training a cross-modal embedding space first. Intrinsic evaluation metrics instead are computationally inexpensive to compute, taking only about 45 minutes for the 17 different tasks. The numbers reported in the upcoming figures indicate the average score for all these tasks, for which the scores on

the individual benchmarks are moved to the Appendix Section A.

For all but the final experiments the GraphSAGE algorithm was run on the *thin* nodes in the Cartesius cluster which has a CPU with 64GB of RAM as seen in Table 5.8. The GPU speed-up was considered marginal, while the initial peak memory requirements of the network was around 68GB under the default settings. Most of the memory requirements were accredited to the *walks.txt* file being loaded which happened pre-training. An effort was made to allow the model to be run on the *thin* node of which significantly more nodes were available than the *fat* node that contained 256GB of RAM. The number of random-walks was decreased slightly such that the peak memory requirements came at 63 GB.

Node Type	Number	Cores	CPU	CLOCK	Memory
broadwell	177	32	E5-2697A v4	2.6 GHz	64 GB
thin	1080	24	E5-2690 v3	2.6 GHz	64 GB
thin	540	24	E5-2695 v2	2.4 GHz	64 GB
fat	32	32	E5-4650	2.7 GHz	256 GB
gpu	64	16	E5-2450 v2	2.5 GHz	96 GB
knl	18	64	7230	1.3 GHz	96 GB

Table 5.8: Cartesius computational cluster CPU-node specifications.

During the parameter-tuning phase for each parameter setting the training was being stopped after 12 hours with the intrinsic evaluation benchmarks being run at every epoch. Training a larger amount of epochs was found to consistently lead to increased performance on our evaluation-benchmark, although slightly, a result that is in line with the findings of Hamilton et al. (2017). Therefore always the result of the last epoch is shown here. As the aggregator function greatly affected training-speed, this resulted in a different amount of epochs for each of the aggregator functions. The variability of the training-speed can be seen in Figure 5.24(b). For the final run with the best parameter settings, the algorithm was run for three consecutive days (72 hours) for improved results. As the evaluation of the model took a significant time off the available training-time, the evaluation frequency was decreased to multiple epochs.

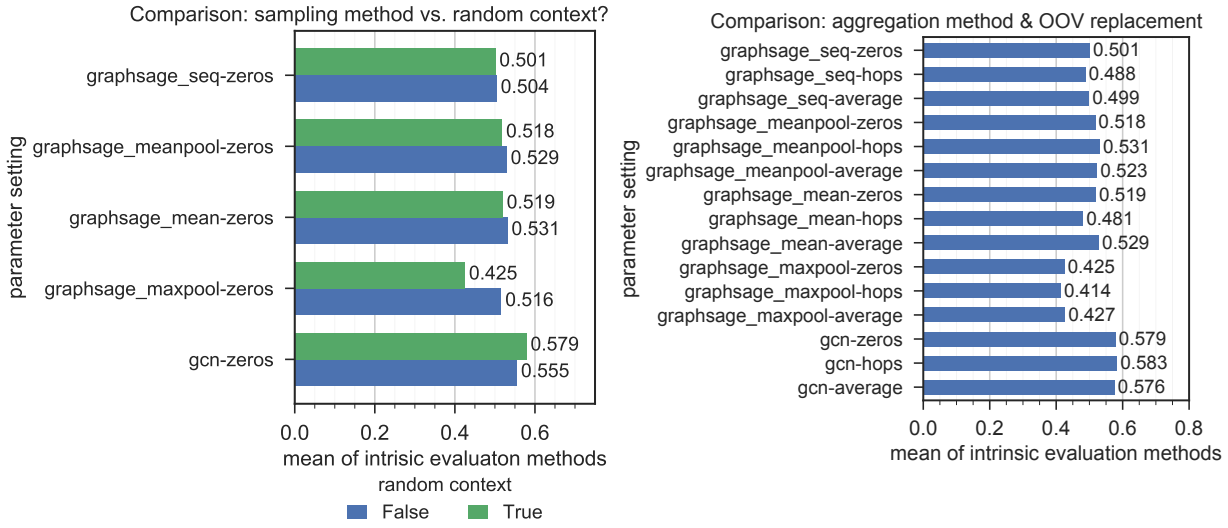
The following parameters were fine-tuned; *max_degree*, *samples_1*, *samples_2*, *neg_sample_size* and *random_context*. A description of these parameters are provided in Table 5.9. See Section 3.3 for a more general explanation of the GraphSAGE algorithm and the work of Hamilton et al. (2017) for more implementation details. Now follows a description of how these parameters were selected.

First, the effect the different aggregator functions had on the intrinsic evaluating benchmarks were calculated in combination with whether they were using random context or not, see Figure 5.22(a). The remainder of the settings were initially left under the default settings. The best settings are used for each

parameter	description
max_degree	For computational efficiency, the maximum node degree was capped at this number of nodes
samples_1	Number of positive examples to consider with 1 hop distance from the node of interest
samples_2	Number of positive examples to consider with 2 hop distance from the node of interest
neg_sample_size	Number of
random_context	Boolean indicating whether for the random walk, random context was used or only direct neighbouring edges
dropout	Dropout rate
model_size	Size of the hidden layer's aggregator function, can be either "large" (1024) or "small" (512)

Table 5.9: GraphSAGE fine-tuned parameters and description.

aggregator function separately. Consequently, the different methods to replace the OOV node-feature embeddings were tested, see Figure 5.22(b). Based on these findings the decision was made to focus most of the remaining efforts towards further tuning the *gcn* and *meanpooling* aggregator functions, with *hops* OOV replacement for gcn and *zeros* for meanpool. Due to the minimal differences between these initialisation methods, the decision was made to choose for a diversification strategy in which a few different aggregator functions and OOV replacement methods were chosen rather than strictly selecting the best performing models. It was still considered probable that the performance on the intrinsic evaluation benchmarks were not an optimal indicator for the performance on the TALL-task.



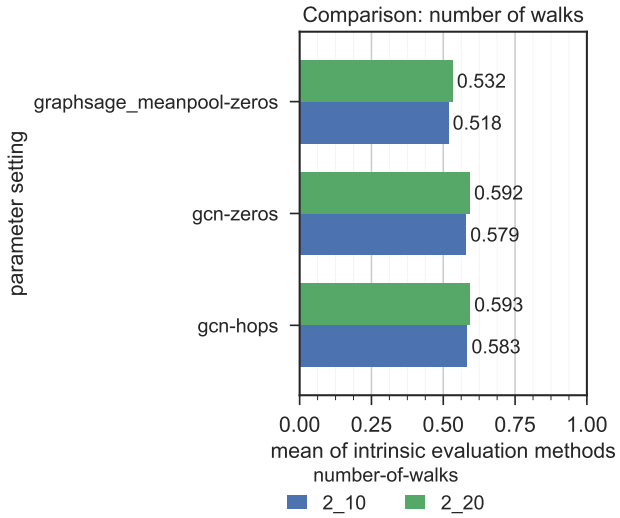
(a) Sampling methods comparison with or without random context

(b) OOV-replacement methods comparison

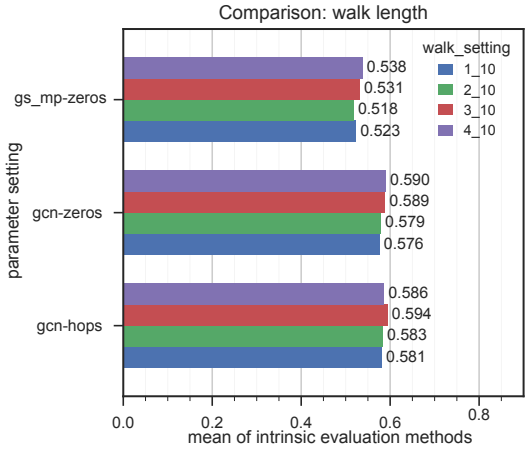
Figure 5.22: Comparisons of the mean of the 17 intrinsic evaluation benchmark scores for the different sampling and OOV-replacement methods used to train GraphSAGE with, part 1.

Due to the edge-sparsity of the ConceptNet graph with most nodes having only one neighbour (Figure 5.18), the decision was made to test only a lower number of walks of 10 or 20, see Figure 5.23(a) as selecting a higher amount of walks would only benefit nodes with significantly more neighbours. In parallel the number of hops for which the amount of information was aggregated was fine-tuned, of which the result is displayed in Figure 5.23(b). Minimal differences between 1 or 2 hops neighbourhood-information aggregation were observed, with more hops slightly lowering the performance. Two hops node-information aggregation was taken. It is important to note that these observations are expected to be significantly

different when the directionality of the edges would be taken into account. As previously discussed, GraphSAGE does not allow for this and therefore it can be expected that a lower number of hops is preferred in order to not include relatively unrelated nodes in a node's local neighbourhood. A more detailed analysis of the differences in performance between these methods is given in the Results Section (6).



(a) Number of walks comparison



(b) Walk length comparison

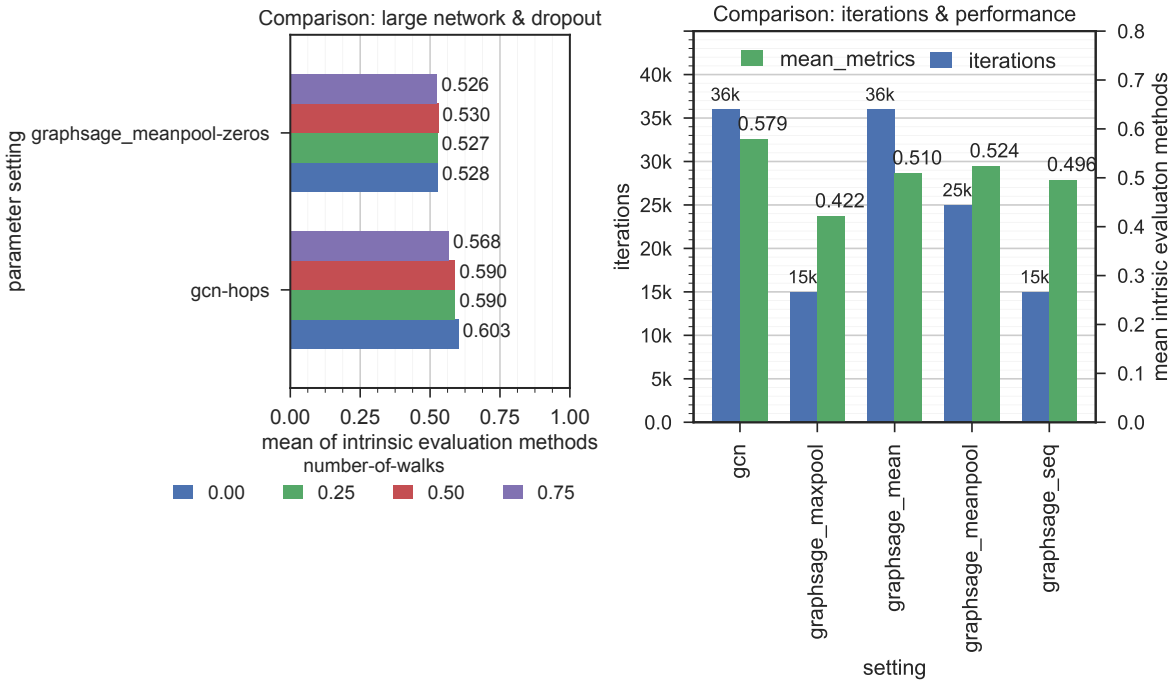
Figure 5.23: Comparisons of the mean of the 17 intrinsic evaluation benchmark scores for a different number of walks and walk lengths used to train GraphSAGE with, part 2.

Lastly, the effect of a larger hidden state size for the selected aggregator functions were tested in combination with dropout, see Figure 5.24(a). No dropout performed best while the difference in performance between a 512 (small) or large (1024) hidden state size made almost no difference. For the final run a walk-length of 2_20 was chosen with *hops* OOV replacements for *gcn* and *zeros* for *meanpool*. Random context was set to False, while the hidden state size of 512 was used. Figure 5.24(b) shows the amount of iterations each aggregator function was able to finish within 12 hours of training with *gcn* and *gcn_meanpool* being the fastest. Despite these differences in performance, it was not found that increased training times of slower aggregator functions resulted in a difference in the ranking of aggregators concerning their performance.

The final run that was trained for 72 hours is reported in the Results Section (6).

5.3 *III - TALL with Embedding Sentence Replacements*

In this experiment the language embeddings obtained in Experiment II were compared to the language embeddings obtained using popular DSM methods in literature. For this, different methods were used to combine the word-level language representation to the sentence level as this is the input of the Gao



(a) Dropout rate comparison

(b) Aggregator training-speed & performance comparison

Figure 5.24: Comparisons of the mean of the 17 intrinsic evaluation benchmark scores for large hidden state and training-speed aggregator function comparison for GraphSAGE, part 3.

et al. (2017) model architecture for the textual domain. This is discussed in Section 5.3.1 and 5.3.2. Thereafter the work of Gao et al. (2017) is reproduced in Section 5.3.3.

Due to the absence of the original work of Gao et al. to include an analysis of the TACoS and Charades-STA dataset, an analysis was included on the vocabulary diversity in Section 5.3.4 and 5.3.5. This was considered especially important as our approach towards obtaining more structured language embeddings assumed that knowledge transfer from the training to test-set domain was of vital importance. As the original intent of Gao et al. was to go from a pre-defined list of classes towards the usage of natural language text, we argue that an appropriate evaluation setup would also *require* an improved representation of language beyond a simple 1-hot encoding of event-classes for high performance. Therefore in Section 5.3.6 we test whether representing a sentence as a 1-hot encoding of words still gives high performance, as this severely limited the transfer of knowledge from the train- to test-set vocabulary.

As Gao et al. made the manipulated TACoS dataset available but did not provide access to their Charades-STA dataset, the decision was made to only focus on the TACoS dataset for the reproduction of their work and comparison with our own language embeddings. However, the textual dataset analysis was still carried out for both datasets.

5.3.1 *Averaging: from Word to Sentence Embeddings*

In Figure 5.26 one can observe an example of the sentence annotations in the TACoS-dataset. As these sentences contained a significant amount of spelling mistakes, the python library

difflib and the function *get_close_matches* was used to find the closest match in the vocabulary of missing words in the vocabulary of the word-embeddings. For this, the sentences were first tokenised with the *nltk* package and each word was lowercased. If no match was found with high confidence, the word was left out. A random sample of the correction proposals with our own vocabulary is shown in Figure 5.25, which was deemed sufficient. Ideally, with perfect overlap between the TACoS language vocabulary and the word-embedding vocabulary, only the ability of the model to align vision and text in the downstream performance TALL-task was tested. After this step, the sentence feature representation was created by averaging the individual word-vector representations. As no significant difference was found between OOV matching or OOV removal, the decision was made to remove the missing words in the vocabulary (Table 6.3). In Table 5.10 one can observe some general statistics of the train and test-set splits as proposed by Gao et al. (2017). One notable detail is that the average sentence length in the validation- and test-set is significantly shorter than in the training-set. As Gao et al. (2017) mention that for longer sentences the performance on the TALL-task is lower, the actual performance on these datasets can be expected to be lower.

```
('weired', 'weirded')
('shildren', 'children')
('staion', 'station')
('4-lane', 'lane')
('Octopus-like', 'octopuslike')
('reinacting', 'resinating')
('siezure', 'seizure')
('fishermans', 'fisherman')
```

Figure 5.25: Random examples of the *difflib* library and the *get_close_matches* function used for correcting OOV spelling mistakes.

```
s13-d21.avi_627_686 The person gets out a knife .
s13-d21.avi_627_686 The person takes out a knife from the drawer .
s13-d21.avi_627_686 He placed the knife on the cutting board .
s13-d21.avi_627_686 The person selects a knife .
```

Figure 5.26: Example of temporal sentence-annotation in the TACoS dataset with multiple alternative sentences per video-segment. The first word consists of the video-name (*s13-d21.avi*) and temporal window (*627_686*) in which the event occurs described in text.

	train	test	validation
# videos	75	25	27
# clips	1604	722	964
# sentences / clip	9.86	10.19	10.32
# words / sentence	6.33	5.65	4.76

Table 5.10: Statistics of the different train- and test-set splits created by Gao et al. (2017).

5.3.2 *Infersent: from Word to Sentence Embeddings*

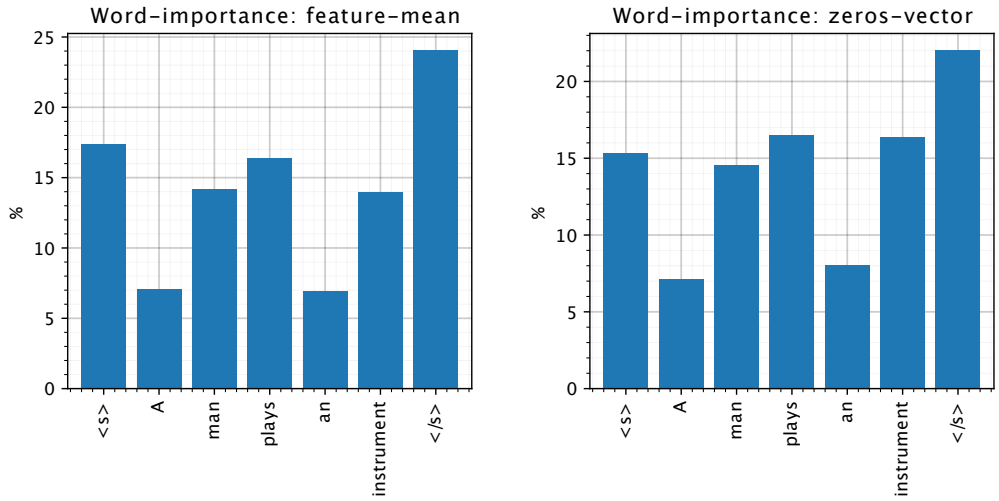
The Infersent algorithm (available on GitHub here) was applied on our word-embeddings to obtain sentence-level language representations. The Infersent algorithm learns the relative importance of the word-embeddings in a supervised matter which results in better sentence representation than a simple word averaging or even SkipThought[3]. The main benefit of InferSent over SkiptThought for our approach is that it (1) uses pre-trained word-embeddings as a starting point which therefore allows the usage of our own language embeddings and (2) has a smaller training-corpus which allows for faster training. SkipThought is trained in an unsupervised matter on 74M sentences from a collection of books whereas InferSent is trained on only 570k sentences from the SNLI corpus that

[3] Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*

contains human-generated English sentence-pairs. In addition, Infersent has shown the capabilities of consistently outperforming SkipThought on a variety of tasks[4].

[4] Conneau et al. (2017)

TRAINING The default settings were used for training Infersent with only changes being made to the hidden state size. The hidden state representation size was changed from 2048 to 2400 in order to obtain the final sentence embedding size of twice this amount of 4800. As our embeddings were obtained using GC on ConceptNet rather than DSM methods, there were no tokens for the start $< s >$ and stop $< /s >$ of sentences. Two attempts were made to add these tokens artificially as they were required for the Infersent algorithm; using word-embedding averages or zero vectors. In Figure 5.27(a) and (b) the importance of the word-embeddings according to the Infersent-model was using the same methodology as used by Conneau et al. (2017). As a lesser importance was preferred for the stop and start tokens, zero vectors (b) were used. The feature average (a) was expected to give higher levels of importance as all dimensions had a relatively high activation, resulting in slightly higher importance levels according to the model.



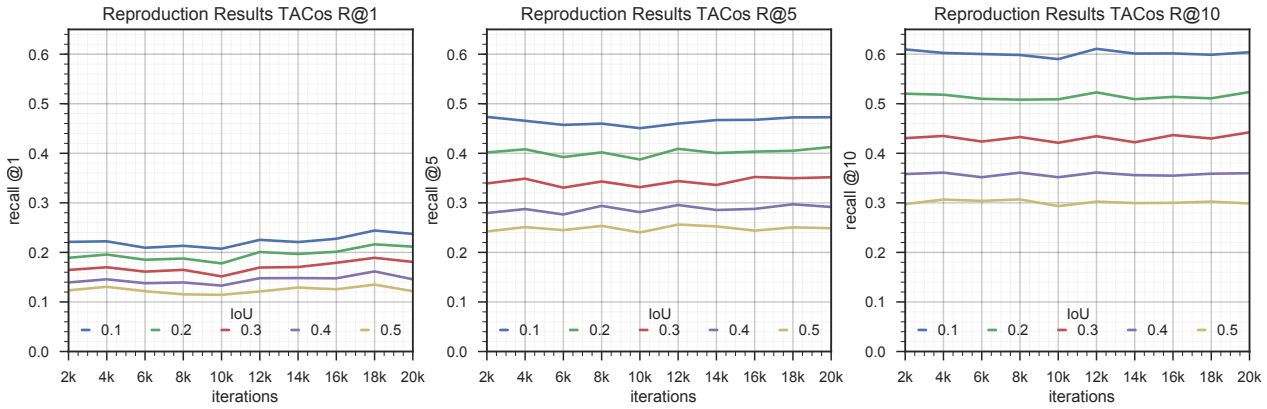
(a) Word embedding-mean for beginning and end of sentence token

(b) Zero-vectors for beginning and end of sentence token

Figure 5.27: Comparison of the relative importance of the beginning and sentence tokens using different initialization methods. (a) word-embedding mean, (b) zero vectors. Notice the different y-scale of both sub-figures.

5·3·3 *TALL Training & Reproduction*

For the TALL training, the default settings were used as it was not expected that different language embeddings would require adjustments in hyper-parameters. The results of Gao et al. (2017) were reproduced and the training behaviour at different IoU, R@n and iterations were analysed (Figure 5.28). From these observations, it was concluded that training converged quickly and was stable with limited risk of over-fitting. Similar but slightly lower performance was obtained as was reported in their paper (Results, 6.



(a) Recall@1 (b) Recall@5 (c) Recall@10

Figure 5.28: Replicating the results of Gao et al. (2017). Training performance is relatively stable after the first 2000 iterations. The default parameters of Gao et al. were used including 20k iterations.

5.3.4 *TACoS Analysis for Zero-Shot Test-Set*

The TACoS train-, validation- and test-set consisted of 100071, 47361, 41628 words respectively. The overlap in vocabulary can be seen in Table 5.11(a). The vocabulary overlap between the train and validation/test dataset is 57.34 and 55.91% respectively. However, this was presumably due to spelling mistakes, when this was compensated for the frequency in which the words were used this resulted in 98.72% and 98.89% overlap for the validation and test-set respectively. With stop-words removed this lead to 97.80% and 98.11% overlap respectively.

5.3.5 *Charades-STA Analysis for Zero-Shot Test-Set*

The TACoS train- and test-set consisted of 89502 and 26922 words respectively. The overlap in vocabulary can be seen in Table 5.11(b). The vocabulary overlap between the train and validation/test dataset is 50.43%. Compensated for the frequency these words were used, this resulted in 99.25% overlap and 98.87% without stop-words. The final results of this experiment are shown in Section 8.

(a)

	train	val	test
train	1556	892	870
val		1199	785
test			1123

(b)

	sentences	videos
train	10146	75
val	4589	27
test	4083	25

Table 5.11: TACoS dataset statistics

(a)

	train	test
train	1150	580
test		870

(b)

	sentences	videos
train	12408	5338
test	3720	1334

Table 5.12: Charades-STA dataset statistics

5.3.6 *One-hot Encoding of Words Alternative*

The use of word embeddings as language representation can be seen as a method to transfer knowledge from the training-set vocabulary to the test-set vocabulary. Given the relatively small datasets in Charades and TACoS and high overlap in vocabu-

lary between the test- and training-set, one question became to which extend this property of the language embeddings was still relevant. Therefore, in this experiment the 1556 unique words in the TACoS training-set were represented as a one-hot encoding of words instead. A sentence was represented as the sum or average of words, with the OOV words still being discarded as was already performed in Section 5.3.1 for a fair comparison. Four different configurations were tested. With or without stop-words and with or without normalisation to unit-length. The results can be observed in Section 6.3.

6 Results & Analysis

The results of Experiment I, II and III are shown below. An interpretation of the results is given in the Discussion Section (7).

6.1 I - Zero-shot Results of Cross-Modal Embedding Space

In Figure 6.1 and 6.2 the mAR is shown as was described in Section 5.1.1 for the cosine and contrastive loss-function. Lower is better. Both figures show the same relationships between the language embedding methods, where Numberbatch > Glove > Lexvec > Word2Vec. The *gcn_** variant of our embeddings were close to Numberbatch.

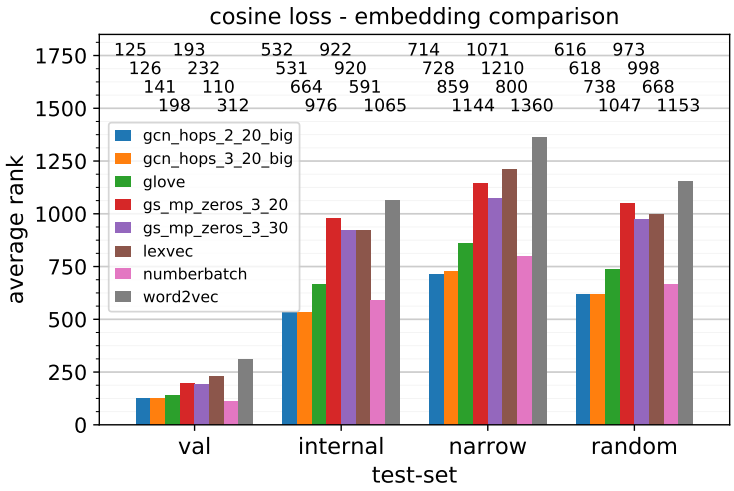


Figure 6.1: Cosine: mAR word-embedding evaluation comparison. Comparison of the zero-shot performance of a variety of word-embedding methods in the task-setting as discussed in Experiment I (4.3). Displaying mAR out of 5579 synsets.

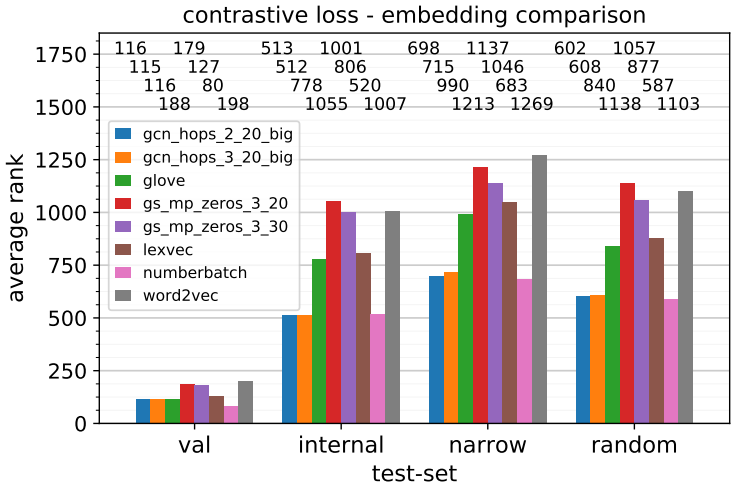


Figure 6.2: Contrastive: mAR word-embedding evaluation comparison. Comparison of the zero-shot performance of a variety of word-embedding methods in the task-setting as discussed in Experiment I (4.3). Displaying mAR out of 5579 synsets.

The zero-shot MAP@10 on the *internal*, *narrow* and *random* is

shown for the cosine and contrastive loss-function in Figure 6.3 and 6.4 respectively. Higher is better. Here one can observe that Numberbatch outperforms DSM language embeddings with a considerably margin except for Glove which is relatively close. Our *gcn_** language embeddings outperforms Numberbatch on all the zero-shot test-sets.

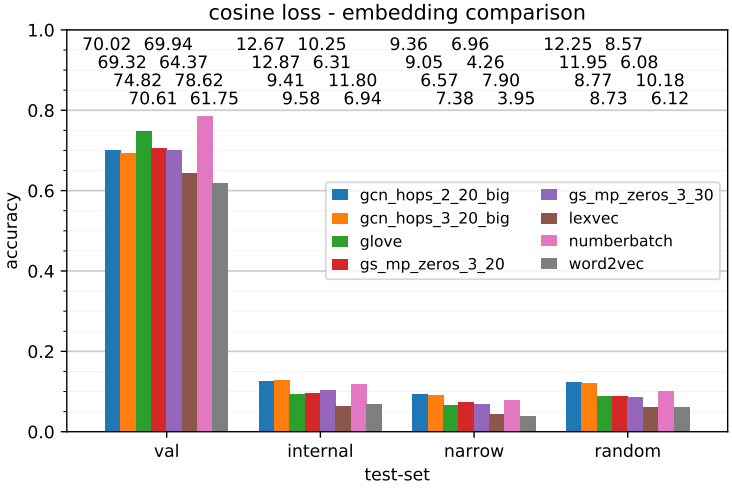


Figure 6.3: Cosine: mAP@10 word-embedding evaluation comparison. Comparison of the zero-shot performance of a variety of word-embedding methods in the task-setting as discussed in Experiment I (4.3). Displaying mAP@10 out of 5579 synsets.

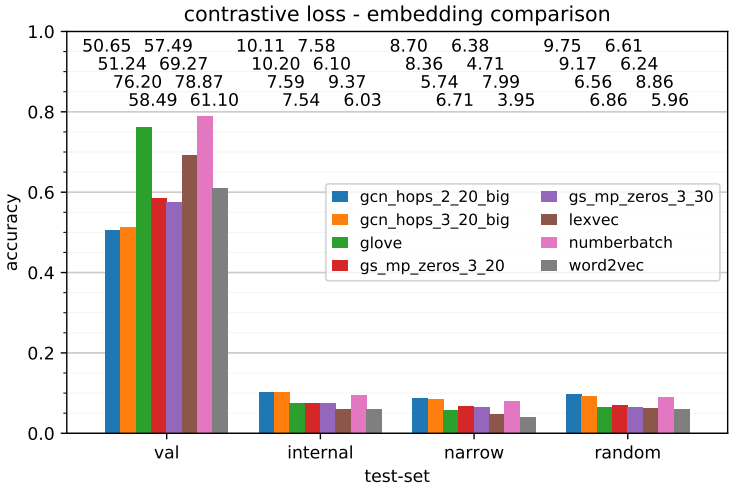


Figure 6.4: Contrastive: mAP@10 word-embedding evaluation comparison. Comparison of the zero-shot performance of a variety of word-embedding methods in the task-setting as discussed in Experiment I (4.3). Displaying mAP@10 out of 5579 synsets.

In order to test whether the results obtained in Fig 1.2 were significant, significant tests were performed. For this, the zero-shot test-classes scores (3183) were aligned for all language embeddings after which the Shapiro test was performed to test if the scores were normally distributed. This was not found to correct for any language embedding pairs for either the contrastive or cosine similarity loss. Therefore, the Wilcoxon test was used to compare whether the two related paired samples came from the same distribution. For all of the embedding pairs, this was not the case see Figure 6.1.

	Cosine Loss (symmetric)				Contrastive Loss (symmetric)			
a ↓, b →	glove	lexvec	numberbatch	word2vec	glove	lexvec	numberbatch	word2vec
glove	—	2.2420e-155	1.6534e-07	6.4403e-159	—	8.9286e-24	4.0647e-82	1.7143e-53
lexvec	—	—	7.130e-163	0.001470	—	—	1.8622e-174	7.6570e-27
numberbatch	—	—	—	2.8535e-163	—	—	—	1.5337e-219
word2vec	—	—	—	—	—	—	—	—

Table 6.1: Statistical significance between the obtained difference in rankings of the popular word-embedding methods as result of Experiment I (4.3).

6.2 *II - GraphSAGE-ConceptNet Embeddings Results*

6.2.1 *Quantitative - Intrinsic Evaluation*

An overview of the 17 intrinsic evaluation scores are shown in Table 6.2. The tasks are divided into the categories; *categorization*, *similarity* and *analogy*. An explanation of the differences can be found in Section 3.2.2. Out of the 17 scores, 5 are the current SOTA including one tie; AP, BLESS, ESSLI_1a, ESSLI_2c (tie), RW. The mean of all scores is given on the right. Here one can observe that despite being first (5) or second (7) highest for most of the tested word-embeddings, on average our score is still significantly lower than most (6 out of 10). This can be solely accredited to the low performance found in analogy-based tasks, which is further discussed in the Discussion 7. The aggregator function applied to combine the local neighbourhood information was found to have a significant impact on the performance of certain intrinsic evaluation tasks. In specific the *mean*-pooling operation keeps certain relationships intact for analogy reasoning while the *gcn* aggregator function greatly degrades the performance on the Google and MSR performance analogy-tasks. An explanation for these differences is given in the Discussion Section (7).

Categorization →	Categorization Tasks						Similarity Tasks								Anology Tasks			
Evaluation →	AP	BLESS	Battig	ESSLI_1a	ESSLI_2b	ESSLI_2c	MEN	MTurk	RG65	RW	SimLex	WS353	WS353R	WS353S	Google	MSR	SemEval	mean ↓
Ev. citation →	Abdulrahman	Baroni	Baroni	Baroni et al.			Bruni	Halawi	Rubenstein	Luong	Hill et	Finkelstein et al.			Mikolov	Mikolov	Jurgens	
Embedding ↓	et al.	et al.	et al.				et al.	et al.	et al.	et al.	al.				et al.	et al.	et al.	
HPCA	0.592	0.680	0.431	0.705	0.750	0.578	0.659	0.606	0.689	0.257	0.275	0.610	0.526	0.708	0.393	0.291	0.152	0.524
NMT	0.415	0.445	0.165	0.568	0.700	0.622	0.492	0.457	0.590	0.306	0.460	0.488	0.444	0.572	0.212	0.434	0.166	0.443
LexVec	0.657	0.845	0.438	0.818	0.750	0.667	0.809	0.712	0.765	0.489	0.419	0.693	0.648	0.754	0.710	0.601	0.187	0.645
Glove	0.637	0.820	0.423	0.750	**0.825**	0.644	0.737	0.633	0.770	0.367	0.371	0.543	0.477	0.662	0.717	0.614	0.170	0.598
morphoRNNLM	0.572	0.605	0.386	0.614	0.775	0.578	0.581	0.620	0.602	0.318	0.242	0.543	0.445	0.645	0.107	0.093	0.175	0.465
PDC	0.639	0.805	0.431	0.818	0.725	0.644	0.773	0.672	0.790	0.472	0.427	0.733	0.673	0.762	**0.748**	0.596	0.190	0.641
Numberbatch	0.724	0.830	**0.472**	0.864	0.750	**0.756**	**0.860**	**0.720**	**0.910**	0.545	**0.651**	**0.755**	**0.687**	**0.824**	0.381	0.539	**0.247**	**0.677**
Word2Vec	0.649	0.805	0.419	0.750	0.800	0.644	0.759	0.682	0.761	0.497	0.442	0.700	0.635	0.772	0.402	**0.712**	0.222	0.627
HDC	0.622	0.815	0.432	0.773	0.750	0.600	0.760	0.658	0.806	0.463	0.407	0.717	0.654	0.768	0.731	0.564	0.199	0.631
FastText	0.632	0.845	0.439	0.773	0.750	0.667	0.764	0.679	0.800	0.479	0.380	0.706	0.655	0.754	0.656	0.521	0.196	0.629
mean →	0.614	0.750	0.404	0.743	0.758	0.640	0.719	0.644	0.748	0.419	0.407	0.649	0.584	0.722	0.506	0.497	0.190	
gcn-hops-2_20-big	0.749	0.850	0.441	0.886	0.750	0.689	0.820	0.699	0.862	0.546	0.507	0.695	0.617	0.808	0.042	0.054	0.164	0.599
gcn-hops-3_20-big	**0.766**	0.870	0.448	0.886	0.725	0.689	0.819	0.699	0.842	**0.553**	0.528	0.719	0.634	0.818	0.041	0.053	0.170	0.604
gcn-hops-3_30-big	0.731	**0.885**	0.440	**0.909**	0.725	**0.756**	0.824	0.694	0.856	0.546	0.508	0.715	0.639	0.796	0.038	0.053	0.170	0.605
gs_mp-zeros-3_20-small	0.649	0.825	0.425	0.773	0.675	0.578	0.728	0.495	0.728	0.363	0.528	0.585	0.444	0.721	0.274	0.276	0.152	0.542
gs_mp-zeros-3_30-small	0.647	0.830	0.430	0.818	0.800	0.600	0.736	0.518	0.758	0.376	0.524	0.578	0.439	0.707	0.260	0.267	0.168	0.556
mean →	0.708	0.852	0.437	0.855	0.735	0.662	0.786	0.621	0.809	0.477	0.519	0.658	0.554	0.770	0.131	0.141	0.165	

Table 6.2: The 17 intrinsic evaluation benchmark scores for all tested word-embedding methods. The tasks can be categorized into categorization, similarity and analogy based tasks.

The intermediate results obtained during parameter-tuning can be found in the appendix; aggregator-OOV-replacement A1, random vs non-random path A2, hops-length vs aggregate function A3, random-walks vs aggregator function A4 and dropout vs aggregate function A5.

6.2.2 *Qualitative Results - TSNe*

The TSNe algorithm was trained on a sample of 20k examples while the vocabulary that was plotted consisted of the set of the first eight sentences parsed to the word-level of the Flickr30k dataset in order to select a diverse set of sentences. First, the dimensionality was reduced from 300 to 10 using PCA, with the sole purpose of improving the computational efficiency of the TSNe algorithm. Ideally, the TSNe plots would have been obtained by using all data-points for the most legitimate comparison, but increasing the number of points above the used 20k sample lead to no progress in the *sklearn* TSNe plotting function. The differences can be observed in Figure 6.5, 6.6 and 6.7 for Glove, Numberbatch and our own embeddings respectively. The colours were only used to indicate the different data-points and have no other meaning.

One can observe that Glove has many concepts clustered around each other, whereas in Numberbatch the concepts are slightly more spread. Our embeddings are spread the most with function words, e.g. *e.g. for, of, out, with, an* being clearly separated from the rest of the vocabulary. In the Numberbatch embeddings, these are all tightly clustered together. Clusters in Numberbatch are mostly related to synonyms or words used in similar contexts; our embeddings seem to be less systematic in the projection of these relationships. For example, *shirt* $\leftrightarrow$ *jeans* is closely together but distant from *shorts* $\leftrightarrow$ *bikers*, in Numberbatch *biker, bicycle, biker, bikers* and *shorts jeans shirt jeans* are all close to each other. Colours, on the other hand, are clustered in our embeddings while they are more scattered in Numberbatch. These observations give some indications of the difference between the three word-embedding and the possible patterns they use in order to capture meaningful relations.

6.2.3 *Flickr30k Zero-Shot Evaluation Analysis*

In Section 5.1.5 an attempt was made to use the Flickr30k dataset to gain a better understanding of the relationships that were being learned in the cross-modal embedding space obtained in Experiment I. For each of the 1000 images that were selected, the image-sentence similarity was calculated and ordered in descending order. On average the correct image-sentence pair occurred at position 467.785 for the obtained *gcn* variant of our language embeddings. As this is only marginally better than random (500), it was not expected that further knowledge could be obtained analysing the performance between different POS-tags.

In Figure 6.8 one can observe the distribution of cosine-distance scores on the word (a) and sentence level (b) for only corresponding sentence-image pairs. In Figure 6.8(c) the frequency is shown that the rank of the corresponding image-sentence pair was observed at for all 1000 sentences (467.785

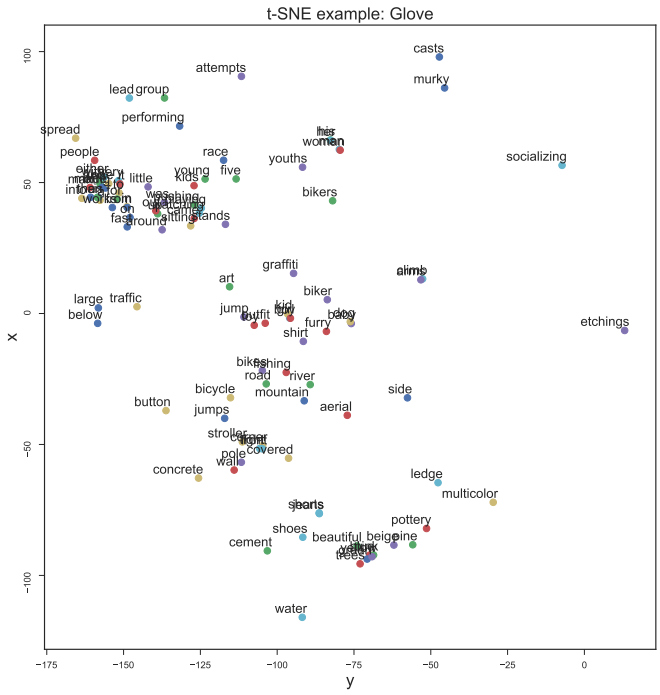


Figure 6.5: Glove TSN-e results. First with PCA the dimensionality was reduced to 10, after which 20k words were used to learn the 2D projections using TSN-e. The same vocabulary was visualized for Figure 6.5, 6.6 and 6.7.

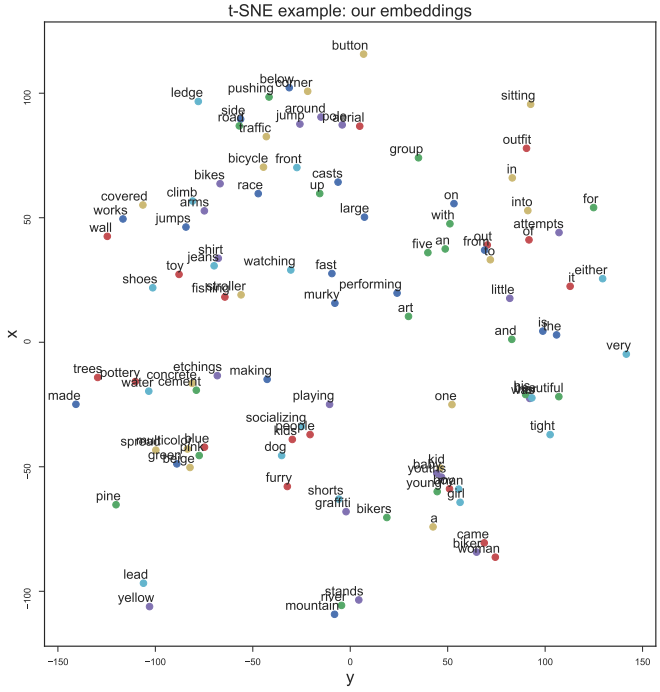


Figure 6.6: Our embeddings gcn-hops-2_20-big TSN-e results. First with PCA the dimensionality was reduced to 10, after which 20k words were used to learn the 2D projections using TSN-e. The same vocabulary was visualized for Figure 6.5, 6.6 and 6.7.

for nouns, 466.285 average). As explained, the image-sentence similarity was calculated by averaging the cosine similarity on the image-word level for each image. Either all words were included or only nouns, however, no significant differences were observed between the two. A visual example of the actual dataset combined with the POS-tags and cosine distance scores between the image-word pairs is shown in Figure 6.9. One can observe that the used NLTK POS-tagger makes mistakes and gives for example the N (noun) tag to adjectives (furry, beige,

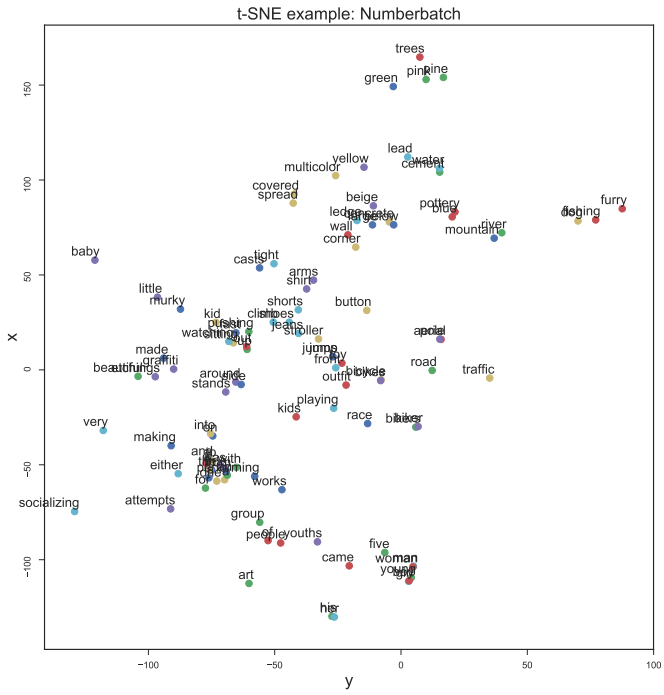
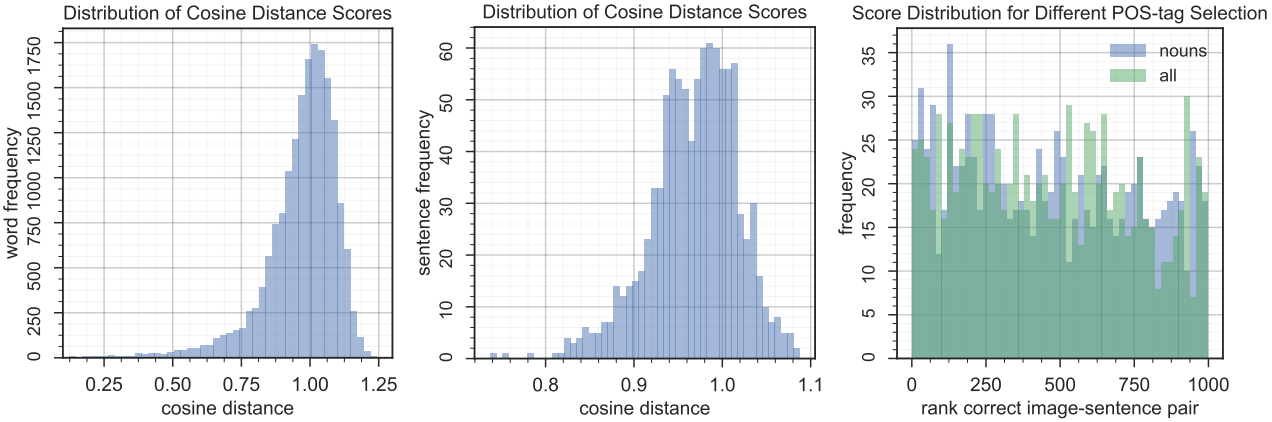


Figure 6.7: Numberbatch TSN-e results. First with PCA the dimensionality was reduced to 10, after which 20k words were used to learn the 2D projections using TSN-e. The same vocabulary was visualized for Figure 6.5, 6.6 and 6.7.

murky).



(a) Distribution of the cosine distances between the corresponding image-word pairs only.

(b) Distribution of the cosine distances between the corresponding image-sentence pairs only. Obtained by averaging the image-word cosine distance for each image-description.

(c) Distribution of ranks that corresponding image-sentence pair are given to when including also 999 non-corresponding image-sentence pairs.

Figure 6.8: Distributions of similarity scores between our *gcn* language embeddings and extracted visual feature vectors (a) and (b). On the right (c), the ranked image-sentence pairs based on similarity.

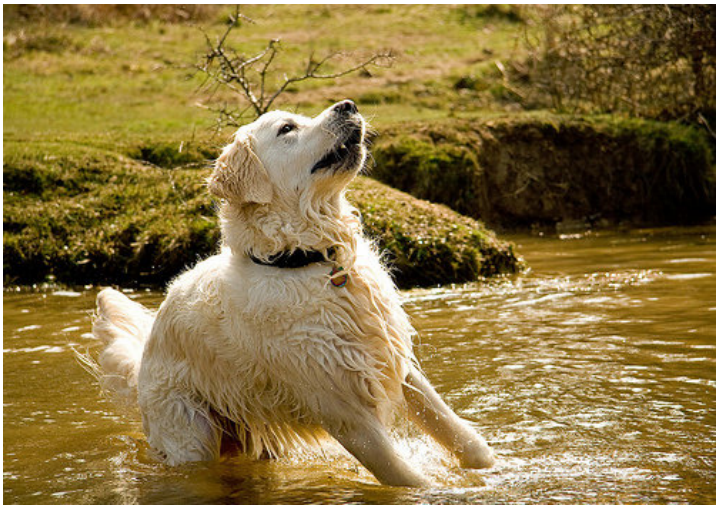


Figure 6.9: Illustration of the cosine distance that the cross-modal embedding space obtained in Experiment I gives to the *3411595210.jpg* image of the Flickr30k dataset to each individual word. An average cosine distance of 0.96 is obtained on the sentence-level (by averaging). The distribution of all image and sentence similarity scores on the word- and sentence-level is shown in Figure 6.8a and b respectively. This method was used to obtain more qualitative knowledge of the relationships that were easily accessible in the cross-modal embedding space in Experiment I by comparing the different POS-tags with their similarity between (non)corresponding image-word pairs. Due to the inability to rank corresponding image-sentence pairs higher than non-corresponding pairs (see Figure 6.8c), any further analysis was not carried out.

6.3 *III - TALL with Sentence Embedding Replacements*

In Table 6.3 the results are shown for the TALL-task as introduced by Gao et al. (2017). The results of the original paper are reproduced from their implementation available on Github and compared with our own language embeddings. As the R@10 scores were not reported in the original paper, these scores are listed as n.a., for completeness these were included for our embeddings and reproduction of their work. There are five different result-groups separated by a horizontal line in Table 6.3 representing a; random baseline, CTRL reproduction and the findings as reported by Gao et al. (2017), our sentence-level Infersent embeddings (GNB-infersent, Section 5.3.2), word-level embedding quality comparison, and lastly the multi-hot word-level sentence embedding results (5.3.6). An underscore indicates the local best performance score whereas in bold the global best is marked.

One can observe that on the sentence-level our approach consistently scored slightly lower than their reported findings in their paper. Our reproduction of their paper also scores lower than their reported findings. A possible explanation for this is that we report the final scores obtained after 20000 iterations, the default setting of their model. With early stopping the results were similar to their reported score but were difficult to justify, as early stopping over multiple runs also sometimes lead to even worse results. Therefore we did not use early-stopping.

As the Infersent algorithm could change the feature repre-

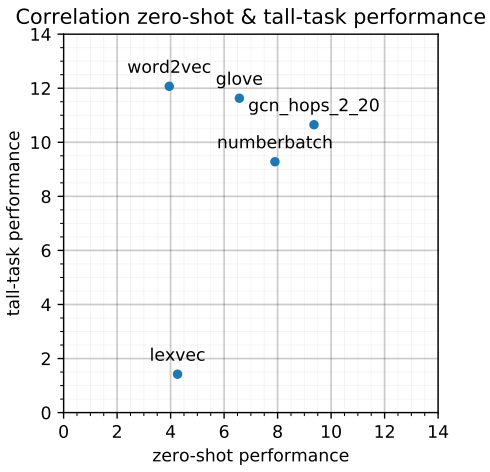


Figure 6.10: Comparison of TALL-task performance of word-embedding methods and zero-shot performance as measured in Experiment I. Used to observe the relationships between the zero-shot performance and the TALL-task performance. Values are corresponding with the ones observed Table 6.3 and Figure 6.3 respectively.

sentation of the words and break-up certain relations, a comparison of the embeddings was also made on the word-level. In Table 6.3 one can observe that the language embeddings obtained with the InferSent algorithm outperforms any of the sentence embeddings obtained by word-embedding averaging. In addition, Word2Vec outperforms the other language-embedding methods obtained on the word-level including our own except when the IoU is just 0.1 for R@5 and R@10. Lastly, in Figure 6.10 one can observe how the zero-shot performance is related to the performance obtained in the TALL-task.

Method	R@1			R@5			R@10		
IoU →	0.5	0.3	0.1	0.5	0.3	0.1	0.5	0.3	0.1
Random (baseline)	0.83	1.81	3.28	3.57	7.03	15.07	n.a	n.a	n.a
CTRL (reg-np) reproduction	12.17	18.07	23.73	24.88	35.17	47.27	**29.88**	**44.23**	**60.40**
CTRL (reg-np) paper	**13.30**	**18.32**	**24.32**	**25.42**	**36.69**	**48.73**	n.a	n.a	n.a
GNB-infersent	12.07	16.73	23.10	24.66	33.75	47.20	28.88	42.27	60.05
glove	11.63	15.53	19.67	23.15	32.11	43.69	28.12	40.68	56.33
word2vec	12.07	16.73	21.82	24.08	33.87	45.82	28.94	41.86	60.23
lexvec	1.42	3.45	4.89	5.19	11.41	22.02	8.87	21.01	36.59
numberbatch	9.28	12.29	18.30	22.75	29.83	44.35	22.75	29.83	44.35
GNB-mean	11.17	15.01	20.06	22.48	31.86	45.06	27.90	41.02	57.48
GNB-GC	10.65	14.91	20.21	21.72	31.08	44.11	27.33	39.92	58.56
GNB-GC-oov-matching	10.97	15.23	19.99	21.72	32.84	47.20	27.75	41.61	**61.28**
one-hot-encoding stopwords	6.89	8.74	12.54	15.77	22.51	34.45	21.85	32.40	50.94
one-hot-encoding no stopwords	6.27	8.74	12.37	15.14	23.02	31.79	20.92	31.97	47.17
one-hot-encoding stopwords normalized	8.16	11.07	15.16	17.14	25.47	37.96	22.95	35.39	51.31
one-hot-encoding no stopwords normalized	6.29	8.13	10.85	13.81	21.45	33.26	19.40	30.81	46.78

Table 6.3: TALL-task performance. Listing the original performance of Gao et al. (2017), our reproduced results, and the performance obtained by substituting their performance with our own sentence language replacement methods. Infersent creates sentence-level embeddings of size 4800, whereas all others create sentence-embeddings using simple word-level averages of dimensionality 300.

7 Discussion

The research questions that were formulated in the introduction are now attempted to be answered. These questions are placed back in their original context and the hypotheses that were formulated in the Introduction (1) are restated. Thereafter, an interpretation is given to the obtained results and concerns are raised about the methodology whenever it was deemed necessary.

7.1 Restating Hypothesis

In the first hypothesis (H1) we formalised our expectations that more structured language embeddings would be beneficial for event-localisation given natural language text due to the increased ability to transfer knowledge from the seen to unseen vocabulary. We argued that the TALL-task is close to a GZSL-task setting, which emphasises the need of transferring knowledge between seen and unseen training-examples. In particular, the large intra-class variety within the visual domain and a large vocabulary-size in the language domain with many complex relationships between words contribute to this need.

In the second hypothesis (H2) we formalised our expectations that language embeddings that contain more visually centred relationships can be better aligned with visual features. We expected that because ConceptNet is centred around objects and the relations they have to other objects and events, this KB could potentially be used to obtain improved language-embeddings for event-localisation due to being more centred around relations in language that have clear visual correspondences (e.g. objects). This is in contrast to the datasets that current distributional language embedding methods use, e.g. Wikipedia, which are less centred around visual-descriptions but more on events described in a historical context.

We designed a number of experiments in order to test whether our approach (1) incorporated our hypothesis and (2) whether this indeed lead to the increase in performance we expected. As our research questions and hypothesis are closely tied together, whether (1) and (2) were indeed observed are discussed jointly in the sections where we attempt to answer RQ1 (7.2.1), RQ2 (7.2.3) and RQ3 (7.2.5). For each of the RQs we also discuss whether the methodology that was used to answer this particular research question was considered sound or whether unforeseen problems were observed that could invalidate our obtained results. For RQ1, RQ2 and RQ3 this is discussed in Section 7.2.2, 7.2.4 and 7.2.6 respectively.

7.2 Relations between Results & RQs

7.2.1 RQ1 - Improved Alignment?

The objective of this research question was to examine whether improved alignment between visual features and language features could be obtained for zero-shot use-cases by creating a more structured representation of language. In specific, whether in accordance with our hypothesis H1 (1.5) adding relational knowledge to distributional semantic language embeddings leads to improved performance in a GZSL task-setting. It was expected that when relational knowledge was added to distributional language embeddings, the added structure could hypothetically improve the transfer of knowledge between seen and unseen visual-textual correspondences.

To answer this research question, we first obtained our own language embeddings in Experiment II using a novel approach that incorporated our hypothesis (H1 & H2 in Section 1.5) and designed in Experiment I a zero-shot evaluation benchmark that tested the extent a variety of language embeddings could be aligned with visual features including unseen textual-visual correspondences. In RQ3 we further explore whether there is an actual correlation between the zero-shot performance scores obtained in Experiment I and the actual TALL-task performance scores, testing our believes whether the TALL-task is actually close to a GZSL problem. In Experiment I we also explored the possibilities to use the obtained cross-modal embedding space in the zero-shot evaluation setup to perform a more qualitative analysis using the Flickr30k dataset. As the results were insignificant and already discussed in Section 6.2.3, this is not further discussed here.

The results of Experiment I (6.1) indicated that approaches that relied upon the combination of relational knowledge with distributional semantics obtained significantly higher zero-shot performance in our evaluation-benchmark (Figure 6.1 and 6.2). The language-embeddings obtained using our approach (9.36%) and Numberbatch (7.90%), which both rely on the structure of ConceptNet, obtained the highest performance on all test-sets including the most difficult *narrow* zero-shot test-set (3rd best was Glove with 6.57%). This was in line with our expectations that were formulated in hypothesis H1. In addition, Numberbatch and the language embeddings obtained using our approach outperformed many of the distributional word-embedding approaches on the intrinsic evaluation benchmarks. This indicates that relying upon relational knowledge and in specific ConceptNet can result in improved general language embedding quality.

The average rank of the corresponding visual-textual features when ranked amongst non-corresponding ones, were shown in Figure 6.1 and 6.2 for a variety of language-embedding

methods. The results indicate that our model using the gcn aggregator with *hops* OOV replacement performs either better or close to Numberbatch and clearly outperforms all other language embedding methods. Considering that the correct corresponding visual-textual feature pairs were ranked on average 698 out of all 5579 synsets, the performance in this experiment was considered mediocre. This indicates that aligning visual and textual features remains difficult even in this simplified setup. Taking the lowest average rank obtained using the two different loss-functions, Glove was closest to our approaches (+20.3%), compared to LexVec (+49.86%) and Word2Vec (+85.67%).

The zero-shot performance calculated as the mAP@10 (Figure 6.3 and 6.4) showed similar relationships between the different language-embeddings as using the mAR evaluation metric. Interestingly, the mAP@10 was relatively high compared to the mAR with our language embeddings obtaining a score of 9.36%. If similar performance was observed at all ranks, a mAR of 106.84 would be observed *(100/9.36*10)* which is significantly lower than the observed 698. This indicates that there is a large difference in the difficulty in which particular synsets could be ranked. Therefore additional testing would be beneficial to investigate what makes certain visual-textual correspondences more difficult to match when compared to others.

To conclude our findings regarding RQ1, we observed that the inclusion of relational knowledge in language embeddings resulted in increased zero-shot performance. In the next section, we highlight possible drawbacks of our employed methodology.

7.2.2 *RQ1 - Remarks about Methodology*

There are a few questions that can be raised about the integrity of the used methodology to answer RQ1. First, we decided to conduct the experiments within the image- rather than the video domain. While we argued that this gave us an advantage by being able to use the ImageNet-hierarchy with a much higher variety of objects and lower the computational cost of our conducted experiments, this does add the concern whether our findings transfer to the video domain. When working in the domain of videos, the recognition of specific motion patterns become important for the classification and localisation of events (Section 2.2.1). De Boer et al. (2017) mention that for zero-shot video event retrieval the availability of mid- and high-level events are contributing more to the obtained performance than the low-level events which contain only objects. Arguably localising mid and high-level events goes beyond the simple recognition of objects, therefore the question remains whether our results obtained using only

object-classes actually generalise to the video domain.

Second, the observed zero-shot performance scores in Experiment I could be unfairly biased towards the language embeddings that relied upon ConceptNet as both ConceptNet and the ImageNet hierarchy (which was used to obtain our zero-shot evaluation benchmark) internally use the WordNet-structure. This shared internal structure could enhance the subsequent matching with visual features artificially. ConceptNet was created by combinnig a variety of KBs, the question therefore becomes how prominently ConceptNet features the WordNet structure. In our experiments, we found that 99.92% of the synsets in our ImageNet-based zero-shot dataset were available in ConceptNet with a high average node-degree of 24.78. This gives some indications that the reliance upon WordNet is strong. Therefore it remains to be seen whether our approach towards testing zero-shot performance is actually reliable. However, as ConceptNet uses a selection of the most popular and largest KBs to date, it is questionable whether this problem could have been prevented.

Lastly, a question that should be posed is whether our obtained word-embeddings were significantly different from the node-feature representation obtained from Numberbatch. Hypothetically, if in our approach no neighbourhood information was aggregated, the node embeddings obtained by GraphSAGE would have been similar to the already SOTA Numberbatch embeddings. Therefore both a qualitative and quantitative analysis was conducted to observe the (dis)similarities between our obtained language embeddings and Numberbatch. In the qualitative analysis, as seen in Figure 6.6 and 6.7, clear differences between our embeddings and theirs is shown in the TSNe plots. Probably the strongest argument that our obtained language embeddings differ from Numberbatch are the significant performance differences obtained in the analogy-based tasks of the intrinsic evaluation benchmarks while performing comparable or better on categorisation and similarity-based tasks. In addition, the cosine distance between the matching vocabulary in our language-embeddings and Numberbatch was plotted in Figure A1. Here one can observe that concepts in our language embeddings with fewer neighbours in ConceptNet have more similarity with Numberbatch (flatter distribution) which is to be expected. However the mode of the cosine distance distribution (Figure A1a) hovers around 1 which indicates that average cosine distance is not similar (0) or dissimilar (2), but somewhere in the middle.

7.2.3 *RQ2 - Zero-shot Dataset?*

The objective in this research question was to obtain a zero-shot dataset that was emphasising the wide-variety of possible events that can be described in natural language text rather

than focusing on correctly classifying fine-grained classes which is frequently the objective in zero-shot evaluation datasets. To answer to which degree this objective was accomplished, two criteria of success were considered. First, whether the zero-shot dataset was showing consistency in the sense that when input-classes from the test-set had more *similarity* with the classes in the training-set, a higher zero-shot performance score was obtained. As this phenomenon is frequently described in the literature (e.g. Deng (2012)) and the objective of zero-shot datasets is to have a disjoint set of classes between the training- and test-set, this was considered a decent measurement of whether an appropriate class-selection technique was used for the test-set. Second, whether the visual-features are coming from a diverse domain which is also close to the vocabulary as seen in natural language text.

To obtain a dataset in accordance with our two criteria of success, the hierarchical structure of ImageNet was selected to have more control over the dissimilarity of the classes in the test-set, while the considerable topic diversity of the images in the ImageNet dataset was expected to be in line with the general nature of events in videos. The downsides of using this dataset were already discussed in Section 7.2.2 and were regarding the different domain this dataset is situated in (images in contrast to videos) which is directly related to the absence of mid- or high-level events that can be observed in videos by the (complex) interaction of objects through time. However, within the video domain and especially the domain of event-localisation, datasets tend to be smaller and less diverse (Section 2.2.2, Figure 2.5) with most datasets not containing a hierarchy in which the events can be related making it difficult to obtain a zero-shot dataset in accordance with our criteria for success within the video domain.

With our ImageNet selection we created three test-datasets; *narrow*, *random* and *internal*, with decreasing average distance between the classes in the training-set and test-sets. In line with our hypothesis we found that the hardest zero-shot dataset was consistently *narrow*, after which *random* and *internal* followed in order (Figure 6.1, 6.2,6.3 and 6.4). Therefore we considered our dataset in accordance with the first criteria for success. The diverse set of image-classes in ImageNet were also considered to be in accordance with the wide variety of objects that could be observed in videos.

7.2.4 *RQ2 - Remarks about Methodology*

Due to time constraints, the parameters that were selected for the cross-modal embedding space obtained in Experiment I were only fine-tuned on Glove after which the same parameter-settings were applied to all language-embeddings. The problem with this approach is that some language embeddings

could benefit from for example early-stopping or extra regularisation to decrease the difference between the validation-set and test-sets performance. However, our results show that a better performance on the validation-set also resulted in a higher performance on the three test-sets (Figure 6.1, 6.2, 6.3 and 6.4), this at least makes it unlikely that a significant performance difference would be obtained by individually optimising the parameters for each individual language-embedding method. However, the possibility can not be ruled out that due to the specific parameter tuning on Glove, the obtained results slightly favour Glove. This is only expected to further strengthen our observation that Numberbatch and our methods outperform Glove in Experiment I.

After the creation of our zero-shot dataset, Xian et al. (2017) publicly released a consistent zero-shot evaluation benchmark using pre-defined splits and a special focus on selecting image-classes in the test-set that were disjoint from the ones in the training-set. Xian et al. showed the effect of different synset-selection strategies on the zero-shot performance, see Figure 7.1. Using synsets from 2 hops away (2H) starting from the training-set synsets resulted in significantly higher zero-shot performance than three hops (3H). In addition, synsets that contained fewer images per synset (L500, L1K, L5K) showed worse performance than the ones with more (M500, M1K, M5K). Therefore, an improvement that could have been made in our task-setup is to use their proposed train- and test-split dataset to directly compare our results with other methods, but more importantly to include synsets that have a variety of images per synset as these synsets were found to be more difficult to classify correctly[1]. In our approach we did not select synsets based on this property and therefore we do not know whether this has a significant impact on the obtained results. However, it was not expected that this specific design decision was introducing bias in the comparison between the different embedding methods as the same test-sets were used for all of them.

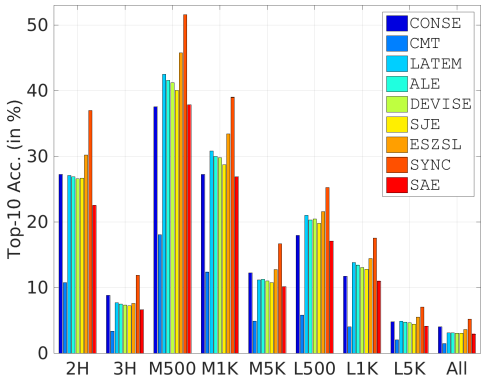


Figure 7.1: Top 10 accuracy of zero-shot performance with different synset-selection strategy and models. Used as an illustration for the large differences in model-performance as result of the synset-selection strategy. Figure reproduced from Xian et al. (2017).

[1] Xian et al. (2017)

7.2.5 *RQ3 - TALL-task Performance?*

In this research question, the objective was to test whether the properties of language embeddings that we hypothesised (H1, H2) to be beneficial for obtaining high performance in the TALL-task actually resulted in high performance. For this, we used the evaluation setup and model-architecture of Gao et al. (2017) to obtain a performance metric on the TALL-task by substituting their representation of language with the language embeddings used in Experiment I.

In Figure 6.10 one can observe the correlation between the zero-shot performance obtained in Experiment I and the performance obtained in the TALL-task in Experiment III. We found

that Numberbatch and our embeddings (gcn_hops_2_20) performed worse on the TALL-task than many of the distributional embedding methods and therefore a negative correlation was observed between the obtained zero-shot performance and TALL-task performance. This was in contrast to our hypothesis that improved zero-shot performance would be beneficial to the TALL-task. Therefore the answer to this research question is that a higher zero-shot performance as obtained in Experiment I is not indicative of improved performance on the task of event-localisation given free-form text queries. However, it should be noted that these results are obtained using the evaluation-setup and model used by Gao et al. (2017). Therefore in Section 5.3.4 and 5.3.5 we analysed in more depth the characteristics of the datasets that were used to evaluate the TALL-task (TACoS and Charades-STA) and in Section 5.3.6 we conducted an experiment to answer whether for this evaluation-setting the transfer of knowledge between known and unknown vocabulary is actually important. As the transfer of knowledge between the train- and test-set is a fundamental requirement in a zero-shot task-setting on which our method is based, this was deemed a crucial step in finding out why a higher zero-shot performance in Experiment I did not transfer to Experiment III. In addition, Gao et al. (2017) introduced the TALL-task as a response to the oversimplification of language in most current event-recognition approaches in which frequently language is represented as a one-hot encoding of event-classes. In this task-setup, there is no transfer of knowledge within the language domain as the classes are entirely disjoint. Therefore arguably the most significant difference between representing language as a one-hot-encoding of event-classes or language-embeddings is precisely the ability to transfer knowledge between words/-classes.

The TACoS and Charades-STA datasets used for training contained 98.11% and 98.87% overlap in vocabulary between their test-sets (Section 5.3.4 and 5.3.5). In addition a small vocabulary size was observed of only 1556 and 1150 words respectively (Table 5.10 and 5.12). These findings question whether given these datasets the transfer of knowledge between words or sentences is actually required. In Table 6.3 one can observe that the differences between a one-hot encoding of words that completely ignores knowledge transfer from the train- to test-set vocabulary, is still performing relatively close to the performance of distributional word-embeddings (Table 6.3, e.g. 17.14 compared to 24.08 for R@5 IoU 0.5). This indicates that knowledge transfer is not particularly important for high performance given this evaluation-setup. This could explain why no clear pattern was found between zero-shot performance and TALL-task performance (Figure 6.10), but remains an open question. At the very least an almost com-

pletely overlapping test- and training-set vocabulary was not deemed realistic. Therefore we conclude that additional testing is required by using a different dataset with a more diverse vocabulary.

Lastly, we obtained SOTA on 5 of the 17 intrinsic evaluation benchmarks and performed second on 7 (Table 6.2) of the ten different popular language embeddings tested. This demonstrates that our method towards obtaining language embeddings succeeds in obtaining general-quality language embeddings.

7.2.6 *RQ3 - Remarks about Methodology*

Arguably to definitively answer RQ3 it is required to test how the different language embedding methods perform under a variety of fundamentally different model architectures designed for the TALL-task. In this work, we solely relied upon the approach of Gao et al. (2017) who introduced the TALL-task and therefore there were no previous methods to compare it to. Ideally, in the future a more comprehensive overview could be created to show how the performance of the different language embedding methods differ under a variety of different modelling choices.

8 Conclusion

In this work, a novel approach was taken towards combining relational knowledge with distributional semantics, in order to obtain improved language embeddings specifically for the task of event-localisation given free-form text queries; the TALL task. Language embeddings obtained by combining both relational knowledge with distributional semantics while emphasising visually-centred relations between words, were hypothesised to improve the alignment with visual-features for this particular task. We argued that the TALL-task is best formalised as a Generalised Zero-shot Learning problem due to the large intra-class variety of images and vocabulary-size within the textual domain, combined with only limited visual-textual correspondences in current datasets to combine the two.

By applying the graph convolution algorithm GraphSAGE on the knowledge base ConceptNet with added distributional node-embedding features, we obtained our own language embeddings in accordance with our hypothesis. The relational knowledge in ConceptNet was expected to lead to more structured language-embeddings and benefit the alignment with visual-features. Whether this indeed leads to improved zero-shot performance was tested on our own zero-shot evaluation-benchmark that emphasised the general nature of events. We observed that language embeddings that featured relational knowledge obtained significantly higher zero-shot performance. However, as both ConceptNet and our evaluation-benchmark relied upon the structure of WordNet, it remains debatable whether this evaluation setup is fair.

SOTA performance was obtained on five popular intrinsic evaluation-benchmarks with competitive results on most others. However, no performance gains were observed on the TALL task using the evaluation-benchmark and model-setup of Gao et al. (2017). We show that under this evaluation-setup, high performance could still be obtained using a multi-hot representation of words due to the small vocabulary-size and roughly 98% overlap between the test- and training-set vocabulary. As this indicates that knowledge transfer between the training- and test-set vocabulary is not required for decent performance, we argue that this evaluation-setup is to a large extent artificial and suggest that more testing is needed on a more diverse dataset to definitively answer whether our obtained language embeddings improve performance on the TALL-task. Nonetheless, we believe that our results show that our methodology is successful in obtaining general purpose language embeddings with many possible extensions for future improvements.

8.1 Future Work

More research could be conducted towards further improving the language embeddings obtained using our approach on intrinsic evaluation benchmarks. In specific, the large amount of OOV words caused by the mismatch between the vocabulary of ConceptNet and popular DSM methods could be improved upon. In the work of Numberbatch already a solution was proposed towards this challenge[1]. Another method to improve the embeddings obtained using our approach is to improve the modelling capabilities of GraphSAGE. Recently, Schlichtkrull et al. (2017) showed that edge direction and relationship type could be taken into account using graph convolutions. It can be expected that significant performance gains can be obtained mainly on the analogy tasks in intrinsic evaluation benchmarks once these properties can also be added to the unsupervised version of GraphSAGE.

[1] Speer et al. (2017)

The combination of fast training times of GCNs, their ability to be trained on CPU rather than GPU without significantly increased training-times and the self-consistency loss-function of unsupervised GCNs approaches, could potentially be exploited in an attempt to learn the representation of language and vision jointly. Xu et al. (2017) stress that one of the major drawbacks of current approaches in event-localisation is that the representations of vision and language are currently fixed by extracting features from networks trained on a different task. With current DSM methods, learning a representation of language is a time-intensive task and requires large quantities of data. This arguably makes it difficult to experiment with different model-architectures to learn the representation of language and vision jointly. Now that we showed that high-quality word-embeddings could be obtained using GraphSAGE with pre-trained distributional node-embedding features using the unsupervised loss-function, further experiments can be conducted to explore whether this can be combined with CNNs in an end-to-end fashion. In our work, we relied upon matching the vocabulary of ConceptNet and language-embedding methods to learn a projection-network that matched the representation of vision to that of language with both feature-representations fixed. However, in future work also the potential can be explored to use this matching directly to optimise both feature representations in an end-to-end fashion. For example, the local-neighbourhood aggregation functions could alter the weight of edges or edge-relations collectively by trying to enforce that the node feature-representation is similar to the visual representation and vice versa. At the same time, the unsupervised loss-function could act as a regularisation method to ensure that the structure of the graph remains intact while also allowing for some flexibility.

In this work, we also argued that the TACoS and Charades-STA datasets are not suitable to evaluate the performance in the TALL-task as almost no emphasis is placed upon relating the seen to unseen vocabulary. Xu et al. (2016) provide a dataset called MSR-VTT designed for video translating to text with emphasis on creating a *large-scale* benchmark dataset. When compared to the TACoS dataset, this dataset is expected to contain more complex visual scenes and a larger number of clip-sentence pairs with more variety between the alternative sentence annotations per clip. This would presumably lead to a more diverse vocabulary while in addition the performance is tested in a much wider domain than is the case in the TACoS or Charades-STA dataset. We recommend to repeat the experiments on the TALL task using this dataset under a variety of different training- and test-set splits due to the still limited size of these datasets.

This work was carried out as part of an internship at QUVA-lab.

This work was carried out on the Dutch national e-infrastructure with the support of SURF Cooperative.

References

Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., and Vijayanarasimhan, S. (2016). Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*.

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Paşca, M., and Soroa, A. (2009). A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics.

Almuhareb, A. (2006). *Attributes in lexical acquisition*. PhD thesis, University of Essex.

Almuhareb, A. and Poesio, M. (2005). Concept learning and categorization from the web. In *proceedings of the annual meeting of the Cognitive Science society*, volume 27.

Bakarov, A. (2018). A survey of word embeddings evaluation methods. *arXiv preprint arXiv:1801.09536*.

Baroni, M., Evert, S., and Lenci, A. (2008). Bridging the gap between semantic theory and computational simulations: Proceedings of the esslli workshop on distributional lexical semantics. *Hamburg, Germany: FOLLI*.

Baroni, M. and Lenci, A. (2010). Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

Baroni, M. and Lenci, A. (2011). How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10. Association for Computational Linguistics.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Bruni, E., Boleda, G., Baroni, M., and Tran, N.-K. (2012). Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 136–145. Association for Computational Linguistics.

Bruni, E., Tram, N., Baroni, M., et al. (2014). Multimodal distributional semantics. *The Journal of Artificial Intelligence Research*, 49:1–47.

Bullinaria, J. A. (2008). Semantic categorization using simple word co-occurrence statistics. In *Proceedings of the ESSLLI Workshop on Distributional Lexical Semantics*, pages 1–8.

Caba Heilbron, F., Carlos Niebles, J., and Ghanem, B. (2016). Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1914–1923.

Caba Heilbron, F., Escorcia, V., Ghanem, B., and Carlos Niebles, J. (2015). Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970.

Cai, H., Zheng, V. W., and Chang, K. C.-C. (2017). A comprehensive survey of graph embedding: Problems, techniques and applications. *arXiv preprint arXiv:1709.07604*.

Carreira, J. and Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. *arXiv preprint arXiv:1705.07750*.

Chiu, B., Korhonen, A., and Pyysalo, S. (2016). Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 1–6.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Dai, X., Singh, B., Zhang, G., Davis, L. S., and Chen, Y. Q. (2017). Temporal context network for

activity localization in videos. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5727–5736. IEEE.

De Boer, M. H., Lu, Y.-J., Zhang, H., Schutte, K., Ngo, C.-W., and Kraaij, W. (2017). Semantic reasoning in zero example video event retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 13(4):60.

Deng, J. (2012). Large scale visual recognition. (Accessed on 08/02/2018).

Deng, J., Berg, A. C., Li, K., and Fei-Fei, L. (2010). What does classifying more than 10,000 image categories tell us? In *European conference on computer vision*, pages 71–84. Springer.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338.

Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. (2009). Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785. IEEE.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2014). Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*.

Faruqui, M., Tsvetkov, Y., Rastogi, P., and Dyer, C. (2016). Problems with evaluation of word embeddings using word similarity tasks. *arXiv preprint arXiv:1605.02276*.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2001). Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2002). Placing search in context: The concept revisited. *ACM Transactions on information systems*, 20(1):116–131.

Fortun, D., Bouthemy, P., and Kervrann, C. (2015). Optical flow modeling and computation: a survey. *Computer Vision and Image Understanding*, 134:1–21.

Fu, Y., Hospedales, T. M., Xiang, T., and Gong, S. (2014). Learning multimodal latent attributes. *IEEE transactions on pattern analysis and machine intelligence*, 36(2):303–316.

Gao, J., Sun, C., Yang, Z., and Nevatia, R. (2017). Tall: Temporal activity localization via language query. *arXiv preprint arXiv:1705.02101*.

Gladkova, A. and Drozd, A. (2016). Intrinsic evaluations of word embeddings: What can we do better? In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 36–42.

Gorban, A., Idrees, H., Jiang, Y.-G., Roshan Zamir, A., Laptev, I., Shah, M., and Sukthankar, R. (2015). THUMOS challenge: Action recognition with a large number of classes. `http://www.thumos.info/`.

Goyal, P. and Ferrara, E. (2017). Graph embedding techniques, applications, and performance: A survey. *arXiv preprint arXiv:1705.02801*.

Gu, C., Sun, C., Vijayanarasimhan, S., Pantofaru, C., Ross, D. A., Toderici, G., Li, Y., Ricco, S., Sukthankar, R., Schmid, C., et al. (2017). Ava: A video dataset of spatio-temporally localized atomic visual actions. *arXiv preprint arXiv:1705.08421*.

Hagberg, A., Swart, P., and S Chult, D. (2008). Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Halawi, G., Dror, G., Gabrilovich, E., and Koren, Y. (2012). Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1406–1414. ACM.

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1025–1035.

Havasi, C., Speer, R., and Alonso, J. (2007). Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In *Recent advances in natural language processing*, pages 27–29. Citeseer.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer.

Hill, F., Cho, K., Jean, S., Devin, C., and Bengio, Y. (2014). Embedding word similarity with neural machine translation. *arXiv preprint arXiv:1412.6448*.

Hill, F., Reichart, R., and Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Huang, C., Change Loy, C., and Tang, X. (2016). Unsupervised learning of discriminative attributes and visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5175–5184.

Idrees, H., Zamir, A. R., Jiang, Y.-G., Gorban, A., Laptev, I., Sukthankar, R., and Shah, M. (2017). The thumos challenge on action recognition for videos âĂIJin the wildâĂİ. *Computer Vision and Image Understanding*, 155:1–23.

Jain, M., van Gemert, J. C., Mensink, T., and Snoek, C. G. (2015a). Objects2action: Classifying and localizing actions without any video example. In *Proceedings of the IEEE international conference on computer vision*, pages 4588–4596.

Jain, M., van Gemert, J. C., and Snoek, C. G. (2015b). What do 15,000 object categories tell us about classifying and localizing actions? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 46–55.

Jastrzebski, S., Leśniak, D., and Czarnecki, W. M. (2017). How to evaluate word embeddings? on importance of data efficiency and simple supervised tasks. *arXiv preprint arXiv:1702.02170*.

Jiang, Y.-G., Liu, J., Roshan Zamir, A., Toderici, G., Laptev, I., Shah, M., and Sukthankar, R. (2014). THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/.

Jones, K. S. and Galliers, J. R. (1995). *Evaluating natural language processing systems: An analysis and review*, volume 1083. Springer Science & Business Media.

Jurgens, D. A., Turney, P. D., Mohammad, S. M., and Holyoak, K. J. (2012). Semeval-2012 task 2: Measuring degrees of relational similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 356–364. Association for Computational Linguistics.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al. (2017). The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Krasin, I., Duerig, T., Alldrin, N., Ferrari, V., Abu-El-Haija, S., Kuznetsova, A., Rom, H., Uijlings, J., Popov, S., Veit, A., Belongie, S., Gomes, V., Gupta, A., Sun, C., Chechik, G., Cai, D., Feng, Z., Narayanan, D., and Murphy, K. (2017). Open-images: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Kuehne, H., Jhuang, H., Stiefelhagen, R., and Serre, T. (2013). Hmdb51: A large video database for human motion recognition. In *High Performance Computing in Science and Engineering âĂŸ12*, pages 571–582. Springer.

Lampert, C. H., Nickisch, H., and Harmeling, S. (2014). Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):453–465.

Lebret, R. and Collobert, R. (2015). " the sum of its parts": Joint learning of word and phrase representations with autoencoders. *arXiv preprint arXiv:1506.05703*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Liu, Y., Zhang, D., Lu, G., and Ma, W.-Y. (2007). A survey of content-based image retrieval with high-level semantics. *Pattern recognition*, 40(1):262–282.

Luong, T., Socher, R., and Manning, C. (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113.

Ma, S., Bargal, S. A., Zhang, J., Sigal, L., and Sclaroff, S. (2017). Do less and achieve more: Training cnns for action recognition utilizing action images from the web. *Pattern Recognition*, 68:334–345.

Mazloom, M., Gavves, E., van de Sande, K., and Snoek, C. (2013). Searching informative concept banks for video event detection. In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*, pages 255–262. ACM.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Miller, G. (1998). *WordNet: An electronic lexical database*. MIT press.

Monfort, M., Zhou, B., Bargal, S. A., Andonian, A., Yan, T., Ramakrishnan, K., Brown, L., Fan, Q., Gutfruend, D., Vondrick, C., et al. (2018). Moments in time dataset: one million videos for event understanding. *arXiv preprint arXiv:1801.03150*.

Nguyen, P., Liu, T., Prasad, G., and Han, B. (2017). Weakly supervised action localization by sparse temporal pooling network. *arXiv preprint arXiv:1712.05080*.

Over, P., Fiscus, J., Joy, D., Michel, M., Awad, G., Kraaij, W., Smeaton, A., QuÃľnot, G., and Ordelman, R. (2015). Trecvid 2015 – an overview of the goals, tasks, data, evaluation mechanisms and metrics.

Palatucci, M., Pomerleau, D., Hinton, G. E., and Mitchell, T. M. (2009). Zero-shot learning with semantic output codes. In *Advances in neural information processing systems*, pages 1410–1418.

Patterson, G. and Hays, J. (2012). Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2751–2758. IEEE.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Pérez, J. S., Meinhardt-Llopis, E., and Facciolo, G. (2013). Tv-l1 optical flow estimation. *Image Processing On Line*, 2013:137–150.

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. (2015). Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 2641–2649. IEEE.

Radinsky, K., Agichtein, E., Gabrilovich, E., and Markovitch, S. (2011). A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pages 337–346. ACM.

Reed, S., Akata, Z., Lee, H., and Schiele, B. (2016). Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 49–58.

Regneri, M., Rohrbach, M., Wetzel, D., Thater, S., Schiele, B., and Pinkal, M. (2013). Grounding action descriptions in videos. *Transactions of the Association of Computational Linguistics*, 1:25–36.

Rohrbach, M., Stark, M., and Schiele, B. (2011). Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1641–1648. IEEE.

Rubenstein, H. and Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Salle, A., Idiart, M., and Villavicencio, A. (2016). Matrix factorization using window sampling and negative sampling for improved word representations. *arXiv preprint arXiv:1606.00819*.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Berg, R. v. d., Titov, I., and Welling, M. (2017). Modeling relational data with graph convolutional networks. *arXiv preprint arXiv:1703.06103*.

Schnabel, T., Labutov, I., Mimno, D., and Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307.

Sigurdsson, G. A., Varol, G., Wang, X., Farhadi, A., Laptev, I., and Gupta, A. (2016). Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Soomro, K., Zamir, A. R., and Shah, M. (2012). Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Speer, R. (2017). Conceptnet numberbatch 17.04: better, less-stereotyped word vectors | conceptnet blog. link. (Accessed on 08/22/2018).

Speer, R., Chin, J., and Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451.

Speer, R. and Havasi, C. (2013). Conceptnet 5: A large semantic network for relational knowledge. In *The PeopleâĂŹs Web Meets NLP*, pages 161–176. Springer.

Speer, R. and Lowry-Duda, J. (2017). Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. *arXiv preprint arXiv:1704.03560*.

Sun, F., Guo, J., Lan, Y., Xu, J., and Cheng, X. (2015). Learning word representations by jointly modeling syntagmatic and paradigmatic relations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 136–145.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 1–9. IEEE.

Wang, H. and Schmid, C. (2013). Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558.

Wang, X. and Gupta, A. (2015). Unsupervised learning of visual representations using videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802.

Wedel, A., Pock, T., Zach, C., Bischof, H., and Cremers, D. (2009). An improved algorithm for tv-l 1 optical flow. In *Statistical and geometrical approaches to visual motion analysis*, pages 23–45. Springer.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. (2010). Caltech-ucsd birds 200.

Xian, Y., Schiele, B., and Akata, Z. (2017). Zero-shot learning-the good, the bad and the ugly. *arXiv preprint arXiv:1703.04394*.

Xie, S., Sun, C., Huang, J., Tu, Z., and Murphy, K. (2017). Rethinking spatiotemporal feature learning for video understanding. *arXiv preprint arXiv:1712.04851*.

Xu, H., Das, A., and Saenko, K. (2017). R-c3d: Region convolutional 3d network for temporal activity detection. *arXiv preprint arXiv:1703.07814*.

Xu, J., Mei, T., Yao, T., and Rui, Y. (2016). Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5288–5296.

Yang, K., Qiao, P., Li, D., Lv, S., and Dou, Y. (2017). Exploring temporal preservation networks for precise temporal action localization. *arXiv preprint arXiv:1708.03280*.

Yang, K., Qiao, P., Li, D., Lv, S., and Dou, Y. (2018). Exploring temporal preservation networks for precise temporal action localization. *CoRR*.

Yuan, J., Ni, B., Yang, X., and Kassim, A. A. (2016). Temporal action localization with pyramid of score distribution features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3093–3102.

Zach, C., Pock, T., and Bischof, H. (2007). A duality based approach for realtime tv-l 1 optical flow. In *Joint Pattern Recognition Symposium*, pages 214–223. Springer.

Zhang, B., Wang, L., Wang, Z., Qiao, Y., and Wang, H. (2016). Real-time action recognition with enhanced motion vector cnns. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2718–2726. IEEE.

Figures

1.1 ConceptNet subgraph. Relations between concepts are shown by arrows and are directional. The text above or below the arrows demonstrate the relationship type (e.g. UsedFor, AtLocation). Relational data from ConceptNet as shown here can potentially be combined with semantic node embedding features to obtain better language embeddings for event-localisation. Figure reproduced from Speer and Havasi (2013). 12

1.2 Comparison between popular language embedding methods on intrinsic evaluation benchmarks and bias-metrics. Figure reproduced from here. In intrinsic evaluation tasks the similarity between word-pairs is calculated based on human judgment and is then compared to the similarity these word-pairs have in the language-embeddings as a measurement for success. 13

2.1 The aim in the TALL-task is to find the temporal boundaries of an event described by a textual description T in video V. (1) A cross-modal embedding space is learned that should give high activation for corresponding V and T. (2) Thereafter a segment proposal network is trained that learns based on the activation output of step (1) to predict the temporal boundaries t_start and t_stop of the event described by T. 19

2.2 A simplified overview of three identified problem-areas of event-localisation in literature. Current approaches can be roughly divided along these three dimensions; *Upper center*: finding a suitable visual representation. *Bottom left*: the computational efficiency in which the localisation and classification of action occurs. *Bottom right*: finding datasets suitable for event-localisation in both size and variety to accomplish this task. The axis are to a certain extend dependend upon each other. 20

2.3 Popular architectures for learning visual video representations that include motion. The models input differ in their representation of time, e.g. on the frame-level (a,c,d) vs. multiple frames (b,e), without (a,b) or with additional motion information (c,d,e). The model architectures also differ in the moment motion information is aggregated, e.g. late (c) vs early fusion (a). Motion patterns can be learned using 3D filters (b,d,e) or 2D approaches (a,c). Figure reproduced from Carreira and Zisserman (2017). 22

2.4 A more in-depth illustration of the approaches towards aggregating temporal information as seen in Figure 2.3. *Single Frame* operates on the frame-level and ignores the temporal aspect. *Late Fusion* compares non-consecutive frames and merges the feature-representation right before prediction. *Early Fusion* takes in n-consecutive frames and learns one joint-representation of time and spatial information that incorporates motion. *Slow Fusion* decreases the temporal-depth in stages while merging and comparing different sub-networks. Figure reproduced from Karpathy et al. (2014). 23

2.5 An overview of a selection of frequently used datasets in CV with emphasis on event recognition and detection. One can observe significant differences in number of classes and samples per class depending on the annotation-level (*action*, *object*, *sentence*, *attribute*) and domain (*image*, *video*). Videos can either be *trimmed* or *untrimmed*, resulting in a classification or localisation task. More on this shown in Figure 2.6 and 2.9. 23
2.6 Properties of datasets. Colors are corresponding with Figure 2.9. Used for further illustrating how methods rely upon different properties of datasets, including the use of KT from the image to video domain. 23
2.7 An example of how events can be seen as a probability over objects. For example the activity BenchPress frequently contains the object bench-press and barbell. Figure reproduced from Jain et al. (2015b). 24
2.8 Instead of relying upon knowledge transfer by using class-to-attribute mappings, Jain et al. (2015a) embed images and textual descriptions of objects in the same space. Thereafter, they extend their zero-shot approach towards localisation of actions in video with promising results. . . 25
2.9 General trends towards which datasets are used for either the localisation or recognition/detection tasks. 27
2.10 Approximate size of the graphs used to apply retrofitting on. Table reproduced by Faruqui et al. (2014) 35

3.1 The TALL module architecture; Cross-modal Temporal Regression Localizer (CTRL) architecture. CTRL contains four modules: a visual encoder to extract features for video clips, a sentence encoder to extract textual embeddings, a multi-modal processing network to generate combined representations for the visual and text domain, and a temporal regression network to produce alignment scores and location offsets. Figure and its description reproduced from Gao et al. (2017). 37
3.2 Significant performance differences were observed using either a learn-able sentence encoder from the word-level or pre-trained sentence-embeddings. Gao et al. (2017) accredit the observed difference due to the limited dataset size that make the training on the word-level unfeasible. Figure reproduced from Gao et al. (2017). 38
3.3 Illustrative schematic overview of the different terms used to extract positive training examples and IoU/nIoL calculations. Figure reproduced from Gao et al. (2017) 38
3.4 Differences in performance obtained on the SimLex intrinsic evaluation benchmark of NMT (yellow), GloVe_300 (blue) and HPCA (green) language embeddings under different percentages of the training data. On the vertical axis is the performance on the task as discussed in the work of Jastrzebski et al. (2017). In supervised-versions of the benchmark (left) one can see that GloVe outperforms the other language embeddings with increased dataset sizes while in the unsupervised version (right) this is not the case. This is used as an argument for using a quantitative metric to denote the ease of KT as an evaluation criterion for language embeddings. Figure reproduced from Jastrzebski et al. (2017). . . . 42
3.5 An example of the method used by Jurgens et al. (2012) to construct the dataset for the anology task; *SemEval*. Based on the human-obtained answers the extend to which the language embeddings are consistent with these findings are used as a measurement for success. 43
3.6 Illustration of how the local neighbourhood of a node is sampled (1) after which the node feature representation is aggregated (2) and an unsupervised loss is applied in (3) that attempts to reconstruct the local neighbourhood. Figure reproduced from Hamilton et al. (2017). 44
3.7 A near linear increase in computational time is observed when the number of edges increases with minimal difference between GPU and CPU time. Figure reproduced from Kipf and Welling (2016). . 45

3.8 Illustration of a graph where all the edges are of the same relation. From a modeling perspective, Hamilton et al. (2017) considers all edge-types equal and without directionality. 45

3.9 Illustration of a graph where there are different edge-relations of which the directionality is important. Many graphs belong into this category, including ConceptNet and ImageNet. 45

3.10 An overview of the training- and testing-time of the different aggregator functions. DW stand for DeepWalk which is one of the benchmark methods used by Hamilton et al. for comparison. Figure reproduced from Hamilton et al. (2017). 46

4.1 An example of how GraphSAGE can be applied upon ConceptNet. W in yellow represents word-embeddings, V represents a visual representation. In red are the verbs and in green are the adjectives. Arrows indicate either undirected (*e.g. related_to*) or directed edges (*used_for*). This figure illustrates that for some concepts there are visual and textual correspondences whereas for others there is not. This possibly allows to relate unrelated to related concepts, ideal for zero-shot use-cases. 51

4.2 Frequency of edge-relationships in ConceptNet. Based on the final selection of concepts from ConceptNet. 53

4.3 A visual example of how the synsets in ImageNet share visual correspondences between more specific examples in the ImageNet-hierarchy. The arrows indicate *is_a* relationships with the more general parent class being on the left-side. Figure reproduced from Deng et al. (2009). . . . 57

5.1 An abstract example of how the nodes were selected from the hierarchy of ImageNet. The N stands for nodes that could be selected in the *narrow* dataset. The R stands for *random* nodes, while I stands for *internal* nodes, whereas X represent the nodes in the training-set. 62

5.2 Example of an xml-entry of the ImageNet tree structure. By nesting synset definitions, the hierarchical structure of ImageNet is obtained. 63

5.3 Illustration of the zero-shot evaluation setup. On the left, the images (V) and text (T) come from the ImageNet dataset where each synset consists of 200 images and the synset name describes the synset in text in possibly multiple alternative ways separated by commas. Features are extracted from the inceptionv1 network and projected down into the language manifold by a MLP. The language manifold is obtained by first matching the synset-name with the word-embedding vocabulary which requires a look-up and filtering operation. The MLP tries to minimise the corresponding projected visual and word-embedding representation of a synset. 64

5.4 Overview of the Inception-v1 architecture with repeating inception modules. An Inception module consists of one or multiple pooling operations (red), convolutions (blue) with in the end a concatenations of the multiple parallel components (green) before connecting to the next Inception-module. For a stronger gradient there are multiple repeated softmax outputs at multiple levels in the network architecture (yellow). 64

5.5 The amount of synsets containing n number of spaces in the whole ImageNet (32297 classes). Used as illustration for difficulty of matching word-embedding vocabulary with ImageNet synset class names. 66

5.6 Matching operations used to match the ImageNet synset names and vocabulary of Word2Vec, LexVec, Numberbatch and Glove. *from_selection* : was used if one of the synset-names was present in all of the word-embedding vocabularies. *multiple* : if all the individual words of one synset-name were present in all language-embedding vocabularies, the synset representation was obtained by averaging the individual embeddings. *redefined* : if a 1-word synonym was found for the synset-descriptions or the class was mapped to a more general parent class. Statistics based on all 5579 synsets. 66
5.7 Schematic overview of the MLP-projection network as was first visualised in Figure 5.3. This figure is best understood in conjunction with the notation introduced in text on the left-hand side. . 67
5.8 Difference between cosine and euclidean distance. 69
5.9 Parameter selection: activation vs. weight initialization method. Results are listed on the validation-set and the 3 different test-sets. 70
5.10 Parameter selection: weight of negative and positive examples. Results are listed on the validation-set and the 3 different test-sets. 71
5.11 Parameter selection: batch normalization (left) and margin of the contrastive loss-function. Results are listed on the validation-set and the 3 different test-sets. 71
5.12 Parameter selection: learning rate. Results are listed on the validation-set and 3 different test-sets. 71
5.13 Parameter selection: dropout. Results are listed on the validation-set and 3 different test-sets. 72
5.14 Final mAP on the validation zero-shot evaluation dataset for both the cosine and contrastive loss-function. 72
5.15 The effect of the missing word replacement from Figure 5.6 on rank@10 and average rank. 73
5.16 Each image in the Flickr30k dataset is described by 5 different people resulting in a diverse sentence-annotated dataset. . . 74
5.17 POS-tag frequency in the used Flickr30k subset. 74
5.18 Distribution of neighbours in our selection of ConceptNet compared with that of Numberbatch. From Numberbatch only the overlapping vocabulary is shown. 76
5.19 The information that is contained in our selection of ConceptNet as a function of the nodes that contain at most n neighbours. In orange one can observe the % of the dataset that is covered by the nodes. 77
5.20 Amount of unique nodes reached starting from with the 5579 synset in our zero-shot dataset and traversing n-hops through the edges in ConceptNet. In the *undirected* case, all edges can be traversed over whereas in the *directed* case the directionality of the edges are taken into account. 78
5.21 Overlap of the vocabulary for our ConceptNet selection using the *undir*ected and *dir*ected dataset as shown in Figure 5.20. Significant vocabulary overlap occurs between the Numberbatch vocabulary and our ConceptNet concept selection. In the end the *undir* ConceptNet concept selection is used. 79
5.22 Comparisons of the mean of the 17 intrinsic evaluation benchmark scores for the different sampling and OOV-replacement methods used to train GraphSAGE with, part 1. 82
5.23 Comparisons of the mean of the 17 intrinsic evaluation benchmark scores for a different number of walks and walk lengths used to train GraphSAGE with, part 2. 83
5.24 Comparisons of the mean of the 17 intrinsic evaluation benchmark scores for large hidden state and training-speed aggregator function comparison for GraphSAGE, part 3. 84
5.25 Random examples of the $difflib$ library and the *get_close_matches* function used for correcting OOV spelling mistakes. . . 85

5.26 Example of temporal sentence-annotation in the TACoS dataset with multiple alternative sentences per video-segment. The first word consists of the video-name (*s13-d21.avi*) and temporal window (*627_686*) in which the event occurs described in text. 85

5.27 Comparison of the relative importance of the beginning and sentence tokens using different initialization methods. (a) word-embedding mean, (b) zero vectors. Notice the different y-scale of both sub-figures. 86

5.28 Replicating the results of Gao et al. (2017). Training performance is relatively stable after the first 2000 iterations. The default parameters of Gao et al. were used including 20k iterations. 87

6.1 Cosine: mAR word-embedding evaluation comparison. Comparison of the zero-shot performance of a variety of word-embedding methods in the task-setting as discussed in Experiment I (4.3). Displaying mAR out of 5579 synsets. 89

6.2 Contrastive: mAR word-embedding evaluation comparison. Comparison of the zero-shot performance of a variety of word-embedding methods in the task-setting as discussed in Experiment I (4.3). Displaying mAR out of 5579 synsets. . . . 89

6.3 Cosine: mAP@10 word-embedding evaluation comparison. Comparison of the zero-shot performance of a variety of word-embedding methods in the task-setting as discussed in Experiment I (4.3). Displaying mAP@10 out of 5579 synsets. 90

6.4 Contrastive: mAP@10 word-embedding evaluation comparison. Comparison of the zero-shot performance of a variety of word-embedding methods in the task-setting as discussed in Experiment I (4.3). Displaying mAP@10 out of 5579 synsets. 90

6.5 Glove TSN-e results. First with PCA the dimensionality was reduced to 10, after which 20k words were used to learn the 2D projections using TSN-e. The same vocabulary was visualized for Figure 6.5, 6.6 and 6.7. 93

6.6 Our embeddings gcn-hops-2_20-big TSN-e results. First with PCA the dimensionality was reduced to 10, after which 20k words were used to learn the 2D projections using TSN-e. The same vocabulary was visualized for Figure 6.5, 6.6 and 6.7. 93

6.7 Numberbatch TSN-e results. First with PCA the dimensionality was reduced to 10, after which 20k words were used to learn the 2D projections using TSN-e. The same vocabulary was visualized for Figure 6.5, 6.6 and 6.7. 94

6.8 Distributions of similarity scores between our *gcn* language embeddings and extracted visual feature vectors (a) and (b). On the right (c), the ranked image-sentence pairs based on similarity. 94

6.9 Illustration of the cosine distance that the cross-modal embedding space obtained in Experiment I gives to the *3411595210.jpg* image of the Flickr30k dataset to each individual word. An average cosine distance of 0.96 is obtained on the sentence-level (by averaging). The distribution of all image and sentence similarity scores on the word- and sentence-level is shown in Figure 6.8a and b respectively. This method was used to obtain more qualitative knowledge of the relationships that were easily accessible in the cross-modal embedding space in Experiment I by comparing the different POS-tags with their similarity between (non)corresponding image-word pairs. Due to the inability to rank corresponding image-sentence pairs higher than non-corresponding pairs (see Figure 6.8c), any further analysis was not carried out. 95

6.10 Comparison of TALL-task performance of word-embedding methods and zero-shot performance as measured in Experiment I. Used to observe the relationships between the zero-shot performance and the TALL-task performance. Values are corresponding with the ones observed Table 6.3 and Figure 6.3 respectively. . . . 95

7.1 Top 10 accuracy of zero-shot performance with different synset-selection strategy and models. Used as an illustration for the large differences in model-performance as result of the synset-selection strategy. Figure reproduced from Xian et al. (2017). 102

A1 Distribution of cosine-similarity scores between corresponding vocabulary pairs of Numberbatch and our embeddings. Percentages each label in the legend represents of the whole distribution (left): all (100%), 1-2 (17.77%), 3-4 (28.75%), 5-8 (29.23%), 9+ (24.25%) 124

Tables

2.1 Summary of major action recognition datasets. 26

5.1 Accuracy and feed-forward + backward time for popular modals using the Pascal Titan X GPU architecture. Benchmarks partially taken from github.com/jcjohnson/cnn-benchmarks. 62

5.2 On the left (a), one can observe the amount of leaf-nodes and internal nodes for ImageNet1k and ImageNet10k. Literal overlap indicates the amount of overlapping synsets, while the Tree-overlap also considers a synset overlapping if a more general synset is available in the ImageNet1k dataset. On the right (b), the original dataset sizes are listed before and after all processing steps as discussed in 5.1.1. The asterisk (*) indicates that this dataset was not included in the test-set directly but were included during the evaluation-setup in order to include random other classes that were not part of either the training- or test-set. 63

5.3 Final settings of the 2-layer MLP. 72

5.4 Example of two raw dataset-entries of the *conceptnet-assertions-5.6.0.csv* version of ConceptNet. 75

5.5 Comparison of the datasets used by Hamilton et al. (2017) with our selection of ConceptNet. 76

5.6 Example of five cleaned entries of the *conceptnet-assertions-5.6.0.csv* version of ConceptNet. 76

5.7 GraphSAGE model input specifications. . 80

5.8 Cartesius computational cluster CPU-node specifications. 81

5.9 GraphSAGE fine-tuned parameters and description. 82

5.10 Statistics of the different train- and test-set splits created by Gao et al. (2017). . . 85

5.11 TACoS dataset statistics 87

5.12 Charades-STA dataset statistics 87

6.1 Statistical significance between the obtained difference in rankings of the popular word-embedding methods as result of Experiment I (4.3). 91

6.2 The 17 intrinsic evaluation benchmark scores for all tested word-embedding methods. The tasks can be categorized into categorization, similarity and analogy based tasks. 91

6.3 TALL-task performance. Listing the original performance of Gao et al. (2017), our reproduced results, and the performance obtained by substituting their performance with our own sentence language replacement methods. Infersent creates sentence-level embeddings of size 4800, whereas all others create sentence-embeddings using simple word-level averages of dimensionality 300. 96

A1 Results of the 17 individual intrinsic evaluation benchmark scores of which the average is displayed in Figure 5.22 (b). As the cosine and contrastive loss-functions did not differ significantly, only the results for the cosine similarity loss is shown. 122

A2 Results of the 17 individual intrinsic evaluation benchmark scores of which the average is displayed in Figure 5.22 (a). . . 122

A3 Results of the 17 individual intrinsic evaluation benchmark scores of which the average is displayed in Figure 5.23 (b). . . 122

A4 Results of the 17 individual intrinsic evaluation benchmark scores of which the average is displayed in Figure 5.23 (a). . . 123

A5 Results of the 17 individual intrinsic evaluation benchmark scores of which the average is displayed in Figure 5.24 (a). . . 123

A6 Sources of the word-embedding methods as listed in Table 6.2 123

Acronyms

AMT Amazon Mechanical Turk 2.3
CNN Convolutional Neural Network 1.1, 2.2, 3.1, 4.3, 8.1
CTRL Cross-modal Temporal Regression Localizer 3.1
CV Computer Vision 2.1, 2.2, 8.1
DL Deep Learning 2.2
DSM Distributed Semantic Model 1.1–1.3, 1.5, 2.0, 3.2, 4.2, 4.4, 5.1–5.3, 6.1, 8.1
GC Graph Convolutions . . . 1.3, 1.7, 2.4, 3.3, 5.3
GCN Graph Convolutional Network . 1.1, 1.3, 2.5, 3.3, 4.2, 8.1
GZSL Generalised Zero-Shot Learning 1.1, 1.3–1.5, 1.7, 3.4, 4.2, 4.3, 4.5, 7.1, 7.2
IoU Intersection over Union 3.1, 5.3, 8.1
KB Knowledge Base 1.1–1.3, 2.3, 2.5, 3.3, 4.2, 5.2, 7.1, 7.2
KT Knowledge Transfer 2.2, 3.2, 8.1
LSTM Long Short Term Memmory 3.1, 3.3
mAP mean Average Precision 4.3, 5.1
mAR mean Average Rank 5.1, 6.1
MLP Multi-Layer Perceptron 5.1, 8.1
nIoL non-Intersection of Length 3.1, 8.1
OOV Out of Vocabulary 5.1–5.3, 7.2, 8.1
PPI Protein to Protein Interaction 3.3
SOTA state-of-the-art 1.3, 1.6, 2.0, 2.2, 2.3, 2.5, 4.5, 5.1, 6.2, 7.2, 8.0
SQ System Query 2.3, 3.1
TALL Temporal Activity Localization via Language 1.1, 1.6–1.8, 2.0–2.4, 3.0–3.2, 3.4, 4.0–4.5, 5.1–5.3, 6.3, 7.1, 7.2, 8.0, 8.1
UID Unique Identifier 4.3, 5.1
UQ User Query 2.3, 3.1
ZSL Zero-Shot Learning 1.1, 4.2

Appendices

A Intrinsic Evaluation Methods Tables

A.1 Aggregator Function vs Feature Initialization

Categorization →	Categorization Tasks						Similarity Tasks								Anology Tasks			
Evaluation →	AP	BLESS	Battig	ESSLI_1a	ESSLI_2b	ESSLI_2c	MEN	MTurk	RG65	RW	SimLex	WS353	WS353R	WS353S	Google	MSR	SemEval	mean ↓
gcn-average	0.669	0.855	**0.448**	**0.886**	0.700	0.622	**0.790**	0.584	**0.853**	**0.537**	0.533	0.693	0.618	0.760	0.029	0.064	0.151	0.576
gcn-hops	0.687	**0.870**	0.424	**0.886**	**0.800**	**0.711**	0.786	0.576	0.844	0.530	0.527	0.682	0.604	0.749	0.033	0.061	0.143	**0.583**
gcn-zeros	**0.699**	0.845	0.426	**0.886**	0.650	**0.711**	0.788	**0.584**	0.852	0.536	0.533	**0.700**	**0.627**	**0.762**	0.029	0.065	0.149	0.579
graphsage_maxpool-average	0.478	0.765	0.343	0.614	0.525	0.489	0.455	0.324	0.725	0.226	0.485	0.448	0.291	0.608	0.160	0.189	0.127	0.427
graphsage_maxpool-hops	0.455	0.705	0.309	0.636	0.475	0.511	0.514	0.339	0.683	0.177	0.441	0.459	0.334	0.615	0.127	0.143	0.106	0.414
graphsage_maxpool-zeros	0.493	0.725	0.330	0.591	0.550	0.467	0.453	0.327	0.728	0.224	0.483	0.457	0.308	0.614	0.164	0.190	0.122	0.425
graphsage_mean-average	0.607	0.820	0.399	**0.886**	0.625	0.556	0.660	0.478	0.836	0.365	0.506	0.544	0.399	0.696	0.199	0.252	**0.159**	0.529
graphsage_mean-hops	0.577	0.845	0.371	0.773	0.525	0.600	0.590	0.429	0.752	0.349	0.480	0.452	0.286	0.608	0.186	0.226	0.126	0.481
graphsage_mean-zeros	0.627	0.805	0.388	0.795	0.525	0.622	0.653	0.468	0.830	0.352	0.522	0.539	0.395	0.694	0.201	0.256	0.154	0.519
graphsage_meanpool-average	0.565	0.845	0.375	0.818	0.525	0.600	0.666	0.459	0.757	0.394	0.527	0.570	0.412	0.685	**0.232**	**0.320**	0.146	0.523
graphsage_meanpool-hops	0.572	0.750	0.388	0.705	0.650	0.644	0.712	0.518	0.718	0.490	**0.573**	0.599	0.452	0.699	0.185	0.233	0.138	0.531
graphsage_meanpool-zeros	0.562	0.805	0.375	0.727	0.525	0.578	0.660	0.478	0.749	0.400	0.557	0.580	0.417	0.693	0.231	0.318	0.146	0.518
graphsage_seq-average	0.515	0.675	0.340	0.636	0.600	0.511	0.602	0.482	0.805	0.426	0.545	0.590	0.481	0.674	0.170	0.283	0.149	0.499
graphsage_seq-hops	0.530	0.730	0.337	0.727	0.575	0.511	0.652	0.482	0.697	0.483	0.516	0.585	0.501	0.654	0.068	0.113	0.131	0.488
graphsage_seq-zeros	0.483	0.685	0.348	0.659	0.575	0.511	0.618	0.484	0.816	0.426	0.559	0.597	0.493	0.675	0.165	0.277	0.147	0.501
mean→	0.568	0.782	0.373	0.748	0.588	0.576	0.640	0.467	0.776	0.394	0.519	0.566	0.441	0.679	0.145	0.199	0.140	

Table A1: Results of the 17 individual intrinsic evaluation benchmark scores of which the average is displayed in Figure 5.22 (b). As the cosine and contrastive loss-functions did not differ significantly, only the results for the cosine similarity loss is shown.

A.2 Random vs Non-Random Path

Comparison with random path set to false. Results can be compared with Table A1 where the path is taken randomly.

Categorization →	Categorization Tasks						Similarity Tasks								Anology Tasks			
Evaluation →	AP	BLESS	Battig	ESSLI_1a	ESSLI_2b	ESSLI_2c	MEN	MTurk	RG65	RW	SimLex	WS353	WS353R	WS353S	Google	MSR	SemEval	mean ↓
gcn-zeros	**0.649**	**0.850**	**0.415**	**0.773**	**0.600**	**0.578**	**0.766**	**0.597**	**0.869**	**0.544**	0.530	**0.673**	**0.579**	**0.754**	0.035	0.063	**0.161**	**0.555**
graphsage_maxpool-zeros	0.520	0.810	0.342	0.727	0.550	0.533	0.690	0.491	0.778	0.499	0.539	0.653	0.514	0.751	0.086	0.160	0.122	0.516
graphsage_mean-zeros	0.535	0.820	0.377	0.682	0.525	0.533	0.716	0.553	0.797	0.526	0.569	0.633	0.508	0.743	0.144	0.227	0.145	0.531
graphsage_meanpool-zeros	0.515	0.745	0.358	0.705	**0.600**	0.556	0.692	0.518	0.782	0.512	0.570	0.653	0.496	0.742	**0.171**	**0.245**	0.140	0.529
graphsage_seq-zeros	0.495	0.640	0.333	0.614	0.550	0.556	0.669	0.490	0.814	0.504	**0.576**	0.627	0.513	0.684	0.131	0.240	0.140	0.504
mean →	0.543	0.773	0.365	0.700	0.565	0.551	0.706	0.530	0.808	0.517	0.557	0.648	0.522	0.735	0.113	0.187	0.141	

Table A2: Results of the 17 individual intrinsic evaluation benchmark scores of which the average is displayed in Figure 5.22 (a).

A.3 Hops Length vs Aggregate Function

Categorization →	Categorization Tasks						Similarity Tasks								Anology Tasks			
Evaluation →	AP	BLESS	Battig	ESSLI_1a	ESSLI_2b	ESSLI_2c	MEN	MTurk	RG65	RW	SimLex	WS353	WS353R	WS353S	Google	MSR	SemEval	mean ↓
gcn-hops-1_10	0.684	0.850	0.425	0.886	0.725	0.667	0.790	0.607	0.852	0.538	0.534	0.684	0.591	0.788	0.038	0.059	0.159	0.581
gcn-hops-2_10	0.687	0.870	0.424	0.886	0.800	0.711	0.786	0.576	0.844	0.530	0.527	0.682	0.604	0.749	0.033	0.061	0.143	0.583
gcn-hops-3_10	0.699	**0.880**	0.420	0.864	**0.875**	0.733	0.798	0.625	**0.885**	0.526	0.511	0.681	0.570	0.771	0.039	0.055	0.167	**0.594**
gcn-hops-4_10	**0.726**	0.875	0.438	**0.909**	0.700	0.689	0.801	0.613	0.883	0.531	0.515	0.686	0.576	0.768	0.043	0.052	0.157	0.586
gcn-zeros-1_10	0.689	0.815	0.422	0.841	0.700	0.600	0.788	0.610	0.830	**0.553**	0.541	**0.712**	**0.627**	0.784	0.033	0.067	**0.179**	0.576
gcn-zeros-2_10	0.699	0.845	0.426	0.886	0.650	0.711	0.788	0.584	0.852	0.536	0.533	0.700	0.627	0.762	0.029	0.065	0.149	0.579
gcn-zeros-3_10	0.709	0.855	**0.446**	0.841	0.725	0.689	0.800	**0.644**	0.864	0.544	0.529	0.700	0.603	**0.799**	0.040	0.065	0.155	0.589
gcn-zeros-4_10	0.694	0.860	0.427	0.864	0.775	**0.756**	**0.802**	0.625	0.883	0.531	0.515	0.685	0.576	0.772	0.036	0.060	0.171	0.590
graphsage_mp-zeros-1_10	0.512	0.795	0.354	0.750	0.650	0.578	0.655	0.475	0.742	0.421	0.556	0.587	0.432	0.707	0.224	0.305	0.148	0.523
graphsage_mp-zeros-2_10	0.562	0.805	0.375	0.727	0.525	0.578	0.660	0.478	0.749	0.400	0.557	0.580	0.417	0.693	**0.231**	**0.318**	0.146	0.518
graphsage_mp-zeros-3_10	0.545	0.805	0.365	0.773	0.575	0.578	0.678	0.482	0.744	0.437	**0.561**	0.622	0.492	0.712	0.216	0.302	0.145	0.531
graphsage_mp-zeros-4_10	0.527	0.805	0.368	0.773	0.700	0.578	0.678	0.482	0.743	0.437	0.560	0.622	0.492	0.711	0.216	0.303	0.145	0.538
AP→	0.644	0.838	0.407	0.833	0.700	0.656	0.752	0.567	0.823	0.499	0.537	0.662	0.551	0.751	0.098	0.143	0.155	

Table A3: Results of the 17 individual intrinsic evaluation benchmark scores of which the average is displayed in Figure 5.23 (b).

A.4 Random Walks Count vs Agregator Function

Categorization →	Categorization Tasks						Similarity Tasks								Anology Tasks			
Evaluation →	AP	BLESS	Battig	ESSLI_1a	ESSLI_2b	ESSLI_2c	MEN	MTurk	RG65	RW	SimLex	WS353	WS353R	WS353S	Google	MSR	SemEval	mean ↓
gcn-hops-2_10	0.687	0.870	0.424	**0.886**	0.800	0.711	0.786	0.576	0.844	0.530	0.527	0.682	0.604	0.749	0.033	0.061	0.143	0.583
gcn-hops-2_20	**0.724**	0.865	**0.435**	0.841	0.750	**0.733**	**0.809**	**0.665**	0.875	0.524	0.520	0.694	0.592	**0.795**	0.038	0.057	0.156	**0.593**
gcn-zeros-2_10	0.699	0.845	0.426	**0.886**	0.650	0.711	0.788	0.584	0.852	**0.536**	0.533	**0.700**	**0.627**	0.762	0.029	0.065	0.149	0.579
gcn-zeros-2_20	0.711	**0.890**	0.418	0.841	**0.850**	0.667	0.798	0.651	**0.875**	0.529	0.517	0.689	0.592	0.777	0.035	0.061	**0.167**	0.592
graphsage_mp-zeros-2_10	0.562	0.805	0.375	0.727	0.525	0.578	0.660	0.478	0.749	0.400	0.557	0.580	0.417	0.693	**0.231**	**0.318**	0.146	0.518
graphsage_mp-zeros-2_20	0.540	0.780	0.373	0.750	0.525	0.578	0.691	0.509	0.776	0.457	**0.567**	0.637	0.495	0.734	0.217	0.282	0.140	0.532
AP→	0.654	0.843	0.408	0.822	0.683	0.663	0.755	0.577	0.829	0.496	0.537	0.664	0.554	0.751	0.097	0.141	0.150	

Table A4: Results of the 17 individual intrinsic evaluation benchmark scores of which the average is displayed in Figure 5.23 (a).

A.5 Dropout vs Aggregate Function

Categorization →	Categorization Tasks						Similarity Tasks								Anology Tasks			
Evaluation →	AP	BLESS	Battig	ESSLI_1a	ESSLI_2b	ESSLI_2c	MEN	MTurk	RG65	RW	SimLex	WS353	WS353R	WS353S	Google	MSR	SemEval	mean ↓
gcn-hops-0.0	**0.751**	0.855	**0.452**	0.841	**0.800**	0.689	**0.814**	**0.676**	**0.867**	**0.542**	0.515	**0.730**	**0.633**	**0.815**	0.041	0.054	**0.172**	0.603
gcn-hops-0.25	**0.751**	**0.875**	0.440	**0.864**	0.725	**0.711**	0.810	0.666	0.864	0.523	0.489	0.690	0.578	0.794	0.037	0.055	0.155	**0.590**
gcn-hops-0.5	0.724	**0.875**	0.435	0.841	0.750	0.667	0.808	0.674	0.864	0.523	0.485	0.715	0.619	0.800	0.037	0.049	0.164	0.590
gcn-hops-0.75	0.701	0.865	0.410	0.841	0.750	0.667	0.798	0.644	0.824	0.489	0.438	0.656	0.564	0.769	0.035	0.042	0.157	0.568
graphsage_mp-zeros-0.0	0.575	0.770	0.385	0.682	0.650	0.511	0.683	0.509	0.724	0.469	**0.571**	0.624	0.472	0.718	0.200	0.281	0.150	0.528
graphsage_mp-zeros-0.25	0.555	0.750	0.392	0.682	0.650	0.533	0.682	0.509	0.720	0.469	0.570	0.625	0.473	0.718	0.200	0.285	0.148	0.527
graphsage_mp-zeros-0.5	0.535	0.735	0.374	0.750	0.650	0.578	0.678	0.509	0.717	0.465	0.570	0.623	0.469	0.715	0.202	0.289	0.148	0.530
graphsage_mp-zeros-0.75	0.552	0.770	0.363	0.750	0.600	0.556	0.668	0.505	0.714	0.456	0.569	0.616	0.455	0.710	**0.205**	**0.298**	0.148	0.526
AP→	0.643	0.812	0.406	0.781	0.697	0.614	0.742	0.587	0.787	0.492	0.526	0.660	0.533	0.755	0.120	0.169	0.155	

Table A5: Results of the 17 individual intrinsic evaluation benchmark scores of which the average is displayed in Figure 5.24 (a).

B Others

B.1 Sentences used for TSNe Visualization

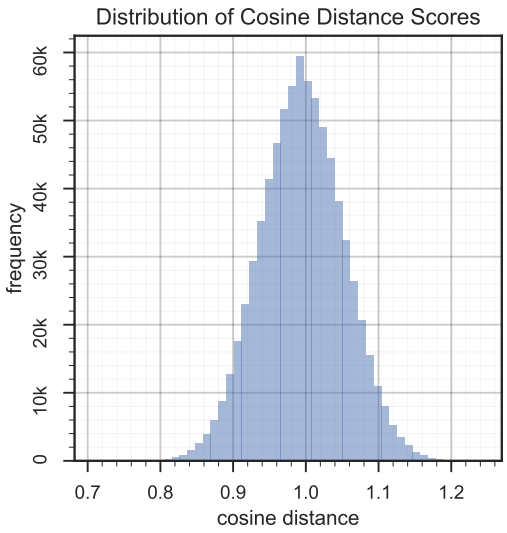
```
'the bikers came around the corner very fast and it was a tight race with the blue biker in the lead',
'a large group of youths sitting and socializing on a cement wall graffiti covered',
'a girl in a green and pink outfit attempts to climb a wall made for kids',
'a little kid in blue shoes is pushing a toy baby in a stroller',
'a man is performing an aerial jump on a bicycle in front of a mountain covered with pine trees',
'a man stands on a concrete ledge and casts his fishing pole into the water below',
'a young boy in a blue shirt and multicolor shorts jumps up out
of the water with his arms spread out to either side',
'five people on bikes in traffic with man watching from the side of the road'
'the furry beige dog is playing in the murky river water'
```

B.2 Word-Embeddings and References

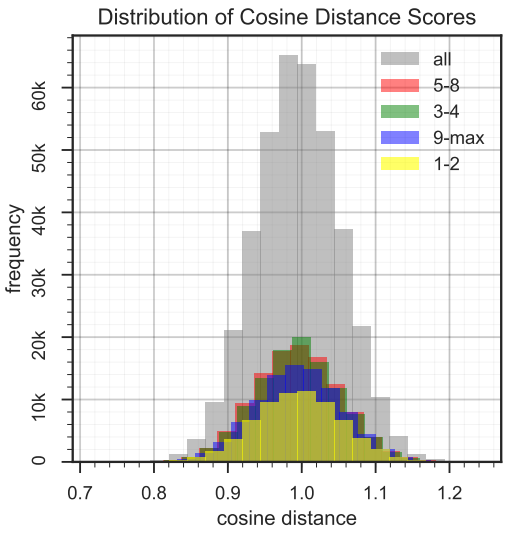
Embedding	Citation
hpca	Lebret et al.
nmt	Hill et al.
lexvec	Salle et al.
glove	Pennington et al.
rnnlm	Luong et al.
pdc	Sun et al.
numberbatch	Speer et al.
word2vec	Mikolov et al.
hdc	Sun et al.
fast_text	Bojanowski et al.

Table A6: Sources of the word-embedding methods as listed in Table 6.2

B.3 *Cosine similarity Numberbatch and Our embeddings.*



(a) All



(b) Separated by amount of neighbours.

Figure A1: Distribution of cosine-similarity scores between corresponding vocabulary pairs of Numberbatch and our embeddings. Percentages each label in the legend represents of the whole distribution (left): all (100%), 1-2 (17.77%), 3-4 (28.75%), 5-8 (29.23%), 9+ (24.25%)