AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

FACULTY OF COMPUTER SCIENCE, ELECTRONICS AND TELECOMMUNICATIONS

DEPARTMENT OF COMPUTER SCIENCE

MASTER THESIS

Word embeddings from lexical ontologies: A comparative study

Author:	*Małgorzata Salawa*
Degree programme:	*Computer Science*
Type of studies:	*Full-time studies*
Supervisor:	*Dr. inż. Aleksander Smywiński-Pohl*
Co-supervisor:	*Prof. António Horta Branco*

Kraków 2019

Uprzedzony(-a) o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2018 r. poz. 1191 z późn. zm.): „Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystycznego wykonania albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.", a także uprzedzony(-a) o odpowiedzialności dyscyplinarnej na podstawie art. 307 ust. 1 ustawy z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce (Dz. U. z 2018 r. poz. 1668, z późn. zm.): „Student podlega odpowiedzialności dyscyplinarnej za naruszenie przepisów obowiązujących w uczelni oraz za czyn uchybiający godności studenta.", oświadczam, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i że nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

Jednocześnie Uczelnia informuje, że zgodnie z art. 15a ww. ustawy o prawie autorskim i prawach pokrewnych Uczelni przysługuje pierwszeństwo w opublikowaniu pracy dyplomowej studenta. Jeżeli Uczelnia nie opublikowała pracy dyplomowej w terminie 6 miesięcy od dnia jej obrony, autor może ją opublikować, chyba że praca jest częścią utworu zbiorowego. Ponadto Uczelnia jako podmiot, o którym mowa w art. 7 ust. 1 pkt 1 ustawy z dnia 20 lipca 2018 r. — Prawo o szkolnictwie wyższym i nauce (Dz. U. z 2018 r. poz. 1668 z późn. zm.), może korzystać bez wynagrodzenia i bez konieczności uzyskania zgody autora z utworu stworzonego przez studenta w wyniku wykonywania obowiązków związanych z odbywaniem studiów, udostępniać utwór ministrowi właściwemu do spraw szkolnictwa wyższego i nauki oraz korzystać z utworów znajdujących się w prowadzonych przez niego bazach danych, w celu sprawdzania z wykorzystaniem systemu antyplagiatowego. Minister właściwy do spraw szkolnictwa wyższego i nauki może korzystać z prac dyplomowych znajdujących się w prowadzonych przez niego bazach danych w zakresie niezbędnym do zapewnienia prawidłowego utrzymania i rozwoju tych baz oraz współpracujących z nimi systemów informatycznych.

Contents

List of Figures

List of Tables

1. Introduction

1.1. Motivation

Neural networks are growing to become the core technology in natural language processing. With this process, a lot of attention is being drawn to the distributional representation of words and their meaning. Such vectorial representations, commonly called word embeddings, aim to encode the lexical and semantic information in the characteristics of the induced vector space.

There exists a multitude of approaches to constructing such representations, differing primarily in the source of the information on lexical semantics, as well as the way it is preserved in the distributional space. A commonly chosen source of the information are large textual corpora, which model the space based on the frequencies of co-occurrence of the words. However, another interesting approach has been studied recently, that is the use of structured, expert-curated lexical graphs, that provide a condensed and precise lexical and semantic information. Such alternative sources are also interesting, as they are backed by psycholinguistical theories regarding the representation of semantics in the human brain. Yet, these sources have not been studied extensively in the research.

1.2. Contribution

This study presents a comparative analysis of word embedding models based on various linguistic sources and obtained using fundamentally different methods. To the best of our knowledge, it is the first systematic analysis of this type, shedding light on different characteristics of the methods, as well as the impact of the chosen lexical source on the performance of the model. The evaluation of the obtained word embeddings comprises both the classical *intrinsic* tasks, such as semantic similarity and relatedness, and the more recent *extrinsic* (or *downstream*) tasks. The latter involve sentence-level processing and classification based on the underlying word embeddings. In such setting, the impact of the use of various models is evaluated through the performance of the top-level sentence-based systems. Such evaluation sheds light on how the lexical information encoded directly in the embedding models is propagated to the higher-level structures of language in the downstream tasks.

2. Semantic representation models

Our driving question is how to represent *the meaning of words*. There exist three broad families of approaches to lexical semantics: semantic networks, feature-based models, and distributional semantic spaces. Each is shortly presented in the following sections.

2.1. Semantic network

The inference-based model of a semantic network was presented by Quillan in 1966 [4]. The nodes in the network represent words (or other lexical units), interlinked with various types of semantic relations as edges. Some types of relations create a strong structure in the network, such as the hierarchy based on the hyponymy relation, that enables inference, e.g. from sentences: 1) *Birds are animals.* 2) *Canary is a bird.*; we can infer (using the transitivity of hyponymy between *animals*, *birds* and *canary*) that 3) *Canary is an animal.* An example of a semantic network is WordNet [5] (further details in Section 3.1.2).

2.2. Feature-based model

The feature-based model presented in 1975 by Minsky [6] and by Bobrow and Norman [7] assumes that the lexical semantics are represented as a map. Each word (concept) is mapped to a list of its *features*. E.g. the word *canary* would have features such as *bird, yellow, sings, has 2 legs, wings,* etc. Such map can also be represented as a network of interconnected lexical units (words and features). An example of a feature-based model is Small World of Words [8] (further details in Section 3.1.2).

2.3. Semantic space

The model of lexical semantics represented as a distributional semantic space was presented in the 1950s by Harris [9] and Osgood et al. [10] and builds upon Wittgenstain's idea that the semantics of a word stems from the context it appears in [11]. As opposed to the other two approaches, it does not represent the words in a graph, but as vectors in a high-dimensional space. Words that are similar or related are represented by vectors *close* to each other, while the vectors of dissimilar and unrelated ones occupy different parts of the space. E.g. some close neighbours of the vector representing a *bird* would

possibly be vectors of *animal, wings, canary, sparrow,* etc., but the distance from a *bird* to *kitchen, computer* or *scarf* would be larger (see Figure 2.1 for a simple example).

Such vectorial representations of words are also called *word embeddings*.

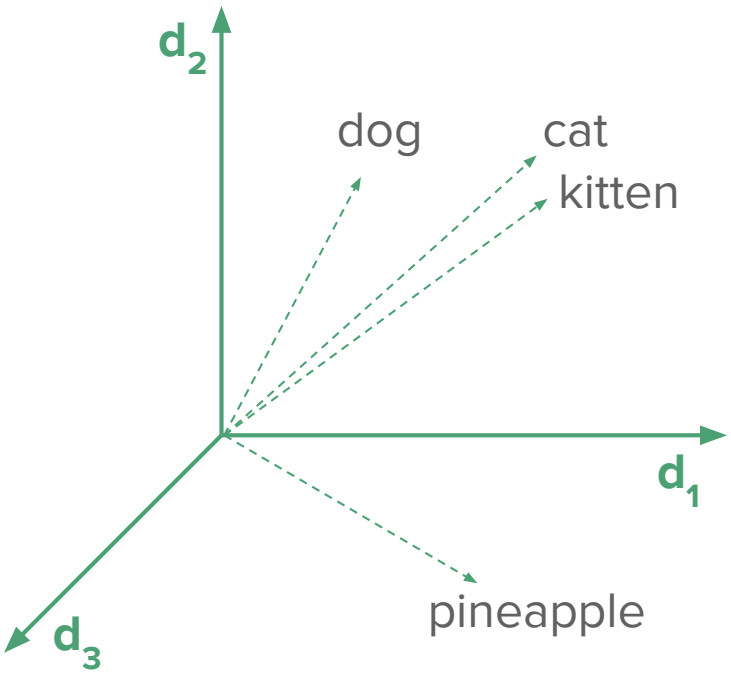

Figure 2.1: Schema of an example 3-dimensional semantic space. The vector representations of words *cat* and *kitten* are very close to each other (high level of similarity), with the vector of *dog* projected further, but still not far (moderate level of similarity), and the vector of *pineapple* projected in a different part of the space (minimal level of similarity).

3. Word embedding

The aim of *embedding*, in general, is to project a set of objects into a vector space in such a way that the relevant properties of the objects are preserved. The common idea is to preserve the similarity of the objects in terms of distance in the embedding space: the similar objects are embedded closer to each other, while dissimilar ones are further away.

Hence, in the light of the definition of a semantic space, the aim of *word embedding* is to project words into a semantic space that approximates the distributional semantic space described in Section 2.3. Such representation enables efficient processing of the words (now represented as vectors of numbers), especially in neural network based systems.

3.1. Word embedding sources

A significant challenge that this process faces is handling the *similarity* between two words. Multiple studies have been exploring the topic of *semantic measures* that could be used to compare, among others, the elements of language: words, sentences, whole documents, as well as concepts defined in knowledge bases [12]. As the authors of [12] note, these measures are based on the analysis of *semantic proxies*, from which semantic evidence can be extracted that will later support comparison of objects. Two broad groups of semantic proxies are *textual corpora* and *lexical ontologies*.

3.1.1. Textual corpora

The semantic measures based on textual corpora rely on the distributional characteristics of natural language, following the assumption that semantically related words tend to co-occur together. This allows for capturing a notion of relatedness between words. For example, since the words *coffee* and *cup* are frequently co-occurring in the corpora, we can expect they are more semantically related than, e.g. words *coffee* and *volcano*, that most probably do not often occur close to each other.

3.1.2. Lexical ontologies

The second group of semantic proxies are the lexical ontologies. These are usually structured knowledge bases, often targeted at a specific domain and curated through experts (e.g. WordNet [5], Gene

POS	Word senses (with sense number)	Gloss
noun	**language#1**, linguistic communication#1	a systematic means of communicating by the use of sounds or conventional symbols
noun	speech#2, speech communication#1, spoken communication#1, spoken language#1, **language#2**, voice communication#1, oral communication#1	(language) communication by word of mouth
noun	lyric#1, words#2, **language#3**	the text of a popular song or musical-comedy number
noun	linguistic process#2, **language#4**	the cognitive processes involved in producing and understanding linguistic communication
noun	**language#5**, speech#8	the mental faculty or power of vocal communication
noun	terminology#1, nomenclature#1, **language#6**	a system of words used to name things in a particular discipline

Table 3.1: Information retrieved from the WordNet database for a query word *language*. Each row represents a synset to which the query word belongs. Each synset is annotated with a part of speech (POS), a gloss (a short description of the synset) and contains a list of word senses together with a *sense number* that allows for the lookup of the exact senses. The remaining data available in WordNet for each synset has been omitted for clarity.

Ontology [13]). The basis for comparison of the objects is the evidence extracted from the *structure* of the ontology.

Two ontologies are explored in the present study: an inference-based semantic network (WordNet), and a feature-based network (Small World of Words).

WordNet[1] is the largest curated lexical semantic network for English [5].[2] Words are grouped into sets of synonyms, called *synsets*, each of which defines a distinct concept. Synsets are connected with each other using conceptual-semantic and lexical relations.

Such structure is built by linguists, therefore all the existing relations are curated by professionals. This makes it a powerful and reliable source of knowledge, both for the users wanting to consult a rich online thesaurus, and for computational linguistics and natural language processing systems.

An example entry from WordNet is presented in Table 3.1.

[1] `https://wordnet.princeton.edu/`

[2] WordNets for many other languages have also been created. A complete list can be found at `http://globalwordnet.org/resources/wordnets-in-the-world/`.

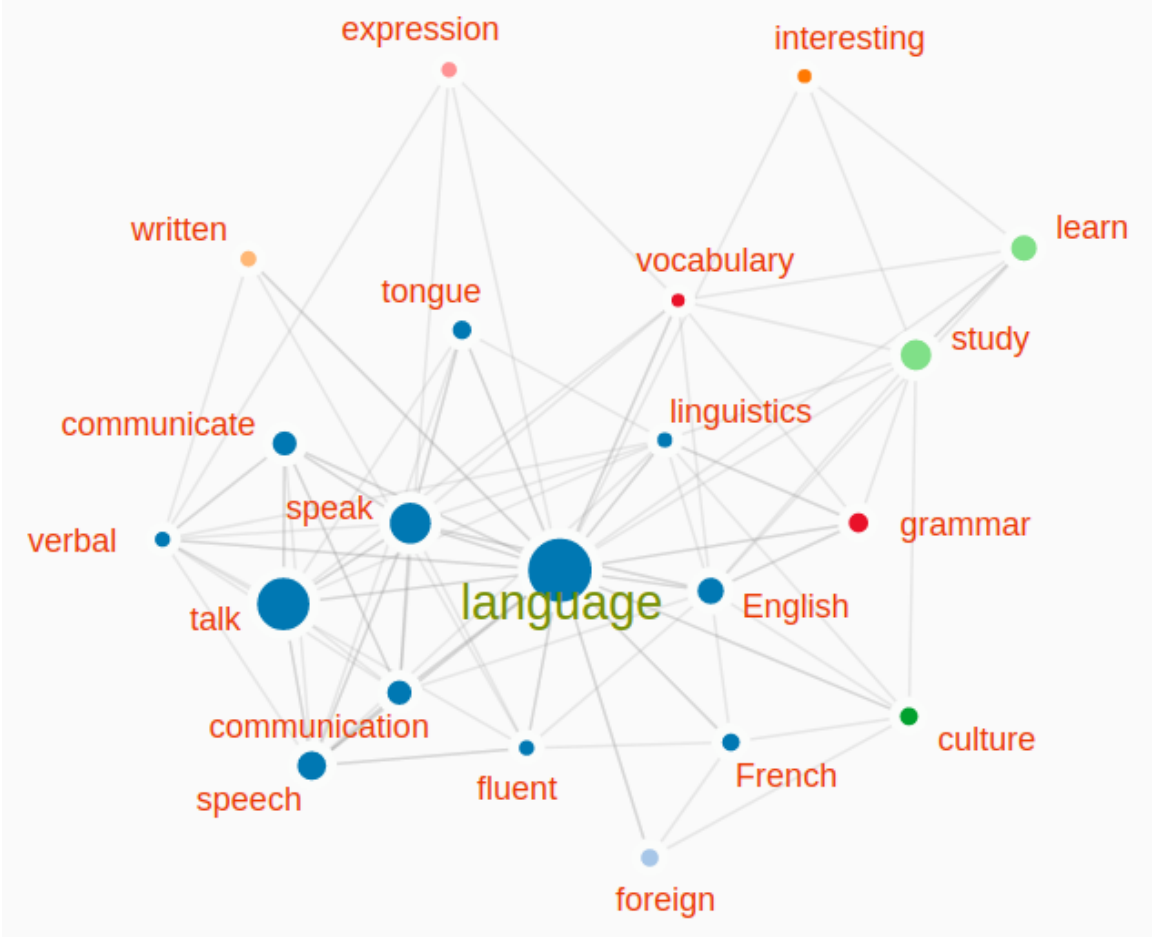

Figure 3.1: Example subgraph of the Small World of Words induced by the query word *language*. The network consists of the words given as responses to the cue *language*. The edges represent the relation of association between the words, i.e. if an edge exists between two nodes, one of the nodes was given as a response to the other. Source: `http://www.smallworldofwords.com/new/visualize`.

Small World of Words[3] (SWOW) is a project aiming to build a mental lexicon in the major languages of the world [8]. The project started with the Dutch language, but has grown since to currently span 14 languages: Dutch, English, Portuguese, German, Spanish (Castilian and Rioplatense), French, Italian, Russian, Japanese, Chinese (Mandarin and Cantonese), Indonesian and Vietnamese.

It is based on a word association task: when presented with a word (the cue), the participant is asked to provide three words that first come to their mind. Participants are allowed to mark a cue as unknown or provide less than three responses by marking it appropriately. An example subgraph of SWOW, presented in Figure 3.1, shows the network induced by the word *language*, where an edge between two words represents the fact that one of them was given as a response to the other.

We use the English dataset described in [8]. It was collected between 2011 and 2018 and consists of over 12000 cue words and responses of over 80000 participants.

[3] `https://www.smallworldofwords.org`

4. Related work

4.1. Word embeddings based on textual corpora

Using neural networks for building a statistical language model and simultaneously training word embeddings using textual corpora was proposed by Bengio et al. in 2003 [14]. The authors presented a feed-forward neural network with an input and projection layer followed by one hidden and an output layer. The network was trained and evaluated on language modelling task using various corpora and showed the superiority of that model over the best existing n-gram models. The model though was computationally expensive because of the large amount of trainable parameters caused by the hidden layer as well as the computation of the softmax function.

The neural word embeddings gained wide popularity thanks to the word2vec model proposed by Mikolov et al. in 2013 [15], [16]. The authors proposed two new architectures (CBOW and Skip-gram), much more efficient due to removal of the hidden layer from the network. The training task was also changed: instead of language modelling (i.e. predicting the next word, given the n preceding context words), the CBOW model tries to predict the middle word, given n context words on the left- and right-hand side, while the Skip-gram model tries to predict the context words given the middle one.

Additionally, in [16] the authors proposed further optimisations to the model. One of them is replacing the *hierarchical softmax* function (an approximation of the full softmax) with an approach called *negative sampling*. This technique avoids the expensive computation of the probabilities distribution over the vocabulary. Instead, for each training sample, *k negative samples* are generated (by choosing the words from the vocabulary randomly or using some defined probabilities), and the error is backpropagated only to the weights of those words, not across the full vocabulary. Another optimisation is *frequent word subsampling*, which reduces the bias towards the frequent words and at the same time reduces the amount of generated training data.

These techniques not only allowed for a significant speedup in the training, but also proved to produce higher-quality embeddings than the model of Bengio et al. [16].

In such models, the similarity of the words is usually computed using the cosine similarity of the respective vectors, using the following formula:

$$\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} = \frac{\sum_{i=1}^{n} \mathbf{a}_i \mathbf{b}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{a}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{b}_i)^2}} \tag{4.1}$$

Word	**Similarity** $\in [0, 1]$
languages	0.8418
vocabulary	0.7185
Language	0.6996
spoken	0.6994
grammar	0.6941
linguistic	0.6868
dialect	0.6806
translation	0.6478
English	0.6374
word	0.6259

Table 4.1: The most similar words to the word *language*, based on vector similarity in the GloVe embedding model [2].

where $\mathbf{a}, \mathbf{b}$ are the vectors, and $\mathbf{a}_i$ denotes the value of the ith coordinate of vector $\mathbf{a}$. Thus, two vectors oriented in the same direction have the cosine similarity of 1, orthogonal vectors have a similarity of 0, and vectors oriented in the opposite directions have a similarity of -1.

An example word with 10 most similar words (with regard to the cosine similarity of the respective vectors) is presented in Table 4.1.

It is worth noting, that only a large textual corpus is needed in order to train such models. The corpus does not require any type of tagging, which is a large advantage of this method over using the ontologies (Section 4.2). Moreover, these models are able to capture the changes in meaning, that occur in the language continuously, by providing the model with the additional training corpora.

It is also noteworthy that all words in the textual corpora are treated as ambiguous, i.e. there is no distinction between different meanings of a word.[1] Since the models rely on the statistical characteristics of co-occurrences of words, the resulting vector representations are usually dominated by a single (most frequent) meaning.

4.2. Word embeddings based on lexical ontologies

Lexical ontologies are represented by graphs, where nodes correspond to the lexical units (e.g. words or synsets in WordNet) connected by the edges that are typed with the lexical relations between them. Thus, obtaining word embeddings from the ontologies comes down to extracting graph node embeddings.

A recent study by Cai et al. [17] presented a comprehensive survey of graph embedding methods. The authors introduce a taxonomy of the methods (based on problem setting, i.e. the type of the input

[1] Unless disambiguation is applied to the corpus before training. However, this is not a common practice due to the vast number of meanings, as well as the difficulty of the disambiguation task.

and output for the algorithm), as well as an outline of five main groups of graph embedding techniques. This study focuses on three of those, that are used most commonly for node embedding: 1) matrix factorisation, 2) random walk, 3) edge reconstruction. These methods represent the graph in saliently different ways, which affects how the properties are preserved in the embedded space. These three groups of methods are presented in the following sections.

4.2.1. Matrix factorisation based methods

These methods represent the graph properties in the form of a matrix, which is then factorised to obtain node embeddings. The main difference lays in how the input matrix is constructed (e.g. adjacency matrix, node proximity matrix) and what objective function they optimise.

As a representative for matrix factorisation based methods, we choose the *Katz index* approach ([18], Eq. 7.63).

The intuition behind this measure is that the larger the number of paths that exist between two nodes, the more similar they are. Therefore, we aim to count all the existing paths between two given nodes. We notice that the result of raising the adjacency matrix M to the power of p is a matrix where each cell m_{ij} represents the number of paths of length p between nodes i and j ([18], Section 6.10). Thus, we can accumulate these counts iteratively:

$$M_G^n = I + \alpha M + \alpha^2 M^2 + \cdots + \alpha^n M^n \tag{4.2}$$

where I is an identity matrix and α is a decay factor, allowing for weighing the influence of longer paths.

Interestingly, if we extend this formula to an infinite sum, following [18] (Section 7.12.4), we can rewrite it in the following way:

$$M_G = \sum_{p=0}^{\infty} (\alpha M)^p = (I - \alpha M)^{-1} \tag{4.3}$$

This allows for simulating an *infinite random walk* on the graph by just manipulating the adjacency matrix, but at the same time bears the cost of a matrix inversion, which is a very computationally expensive operation, especially for larger graphs.

The method was successfully applied to WordNet, where the authors showed the resulting embeddings outperformed the mainstream text-based embeddings in the semantic similarity task [19].

4.2.2. Random walk based methods

These methods represent the graph as a list of random walk paths sampled from the graph, to which some deep learning method is then applied in order to extract node embeddings. A common technique is training a Skip-Gram model over such synthetic corpus, or using recurrent neural networks, such as the ones based on Long-Short-Term Memory (LSTM) units.

The Skip-Gram based method of embedding nodes in a graph was introduced by Perozzi et al. [20] as *DeepWalk* and was used to learn latent representations in social networks. It was later generalised by

WordNet	*singing_voice vocalisation communication language speech dictation speech words* *publication communication language synchronic linguistic_communication* *infix affix word language word palindrome word derivative linguistics*
SWOW	*journalist write script language learning university college* *coin expense usage proper pronunciation linguistic language words many choices dilemma* *meal bread lunch late time wait pause still anyway because then them*

Table 4.2: Examples of the pseudosentences (or their fragments, for clarity) generated using the random walk technique based on WordNet and SWOW.

Grover and Leskovec [21], that allowed for a more flexible notion of neighbourhood between the nodes achieved through biasing the random walk.

Following a similar approach, Goikoetxea et al. [22] applied it to WordNet to obtain word embeddings that proved to outperform or perform comparably to the text-based ones on the semantic similarity task. Additionally the authors show that joining both text- and graph-based embeddings further improved the scores, which suggests that these two models encode different semantic information in the embeddings.

We choose this model as a representative for the random walk based methods and adapt it as needed for comparability (see details in Sections 5.1.2 and 5.2.2). Examples of the pseudosentences generated for both graphs are presented in Table 4.2. It is worth noting, how the ambiguity of the words is visible in the SWOW graph, e.g. in the last example: $\ldots pause \rightarrow still \rightarrow anyway \ldots$, where *still* is associated with two entirely different concepts: a) *pause* (in the sense of lack of movement), and b) *anyway* (in the sense of *nevertheless*). Despite the fact that this does not take place in the sentences generated based on WordNet, as the graph is synset-based, the synset information is not retained in the sentences. Therefore, in the second phase (training the Skim-gram model), all meanings of a given word, corresponding to various synsets that it belongs to, will be encoded in a single vector.

4.2.3. Edge reconstruction based methods

Edge reconstruction based models operate on graphs represented by edge lists. An edge, sometimes also called a *relation*, is a triple $< lhs, rel, rhs >$, where lhs (left-hand-side) and rhs (right-hand-side) are nodes connected by a relation of type rel. The system is trained to differentiate triples that are feasible (present in the graph) from the infeasible ones.

The objective function optimized in the model is either maximizing the edge reconstruction probability or minimizing the edge reconstruction loss. The latter can be further divided into distance-based loss and margin-based ranking loss. Since most of the existing knowledge graph embedding methods choose to optimize margin based ranking loss [17], we choose a method from this subgroup as a representative of the edge reconstruction models.

In these models, the goal is to preserve a ranking of a true triplet $< lhs, rel, rhs >$ over that of a false triplet $< lhs', rel, rhs' >$ that does not exist in the graph. This is achieved by designing an energy function $f_{rel}(lhs, rhs)$, interpreted as a distance between the nodes lhs and rhs in the context of relation rel. Thus, the energy value is lower for the feasible triplets and higher for the infeasible ones. The margin-based ranking loss is defined in general as:

$$O_{rank} = \min \sum_{\substack{<lhs,rel,rhs> \in S \\ <lhs',rel,rhs'> \notin S}} \max(0, \gamma + f_{rel}(lhs, rhs) - f_{rel}(lhs', rhs')) \tag{4.4}$$

Most existing methods use Eq. 4.4 as the objective function, while varying in the choice of the energy function f [17].

Since the dataset based on an edge list consists only of the positive samples (an edge list contains only the existing relations in the graph), systems employ the negative sampling technique in training of the model (the corrupted, negative samples are generated during training).

The representative model chosen for the comparison is *Semantic Matching Energy (SME)* introduced by [23]. This method has already exhibited potential in encoding the underlying structure of WordNet (as noted further).

The SME function is designed as a neural network and based on the intuition that the relation type rel should first be used to extract the semantic information from the nodes. Therefore, the lhs and rhs nodes are first combined separately with rel using a combination function g, which creates new, relation-dependent embeddings of the nodes. The resulting vectors are in a common vector space, thus at this point they can be *matched* against each other. The general scheme of the SME function is shown in Figure 4.1.

The matching can be performed by a complex operation, whose parameters are learned during training, or a simple operator, such as a dot product. The authors opt for the latter, taking into account its simplicity combined with good performance in related research. We follow this choice in our experiments.

Two variants of the combination function g are introduced: *linear* and *bilinear*. In the linear version, the g functions - g_{left} for the left-hand-side context, and g_{right} for the right-hand-side context - are simply linear layers in the network. In the bilinear version, the g functions are more complex and use 3-modes tensors as weights.

The authors also evaluate the potential of the method in encoding the underlying structure of WordNet. They select a subset of words denoting the names of the continents and countries in the world, as well as the US states, given their underlying structure in the ontology (through the meronymy-holonymy hierarchy). Subsequently, they use t-SNE [24] to obtain 2-dimensional visualisations of the raw word embeddings, as well as the relation-dependent embeddings. They conclude that the embeddings obtained using the complex, bilinear variant of SME are more interpretable when used *in the context of a relation type*. This conclusion supported the choice of the simpler, linear version of SME for the experiments in the present study, due to the better interpretability of the word embeddings without any relational context.

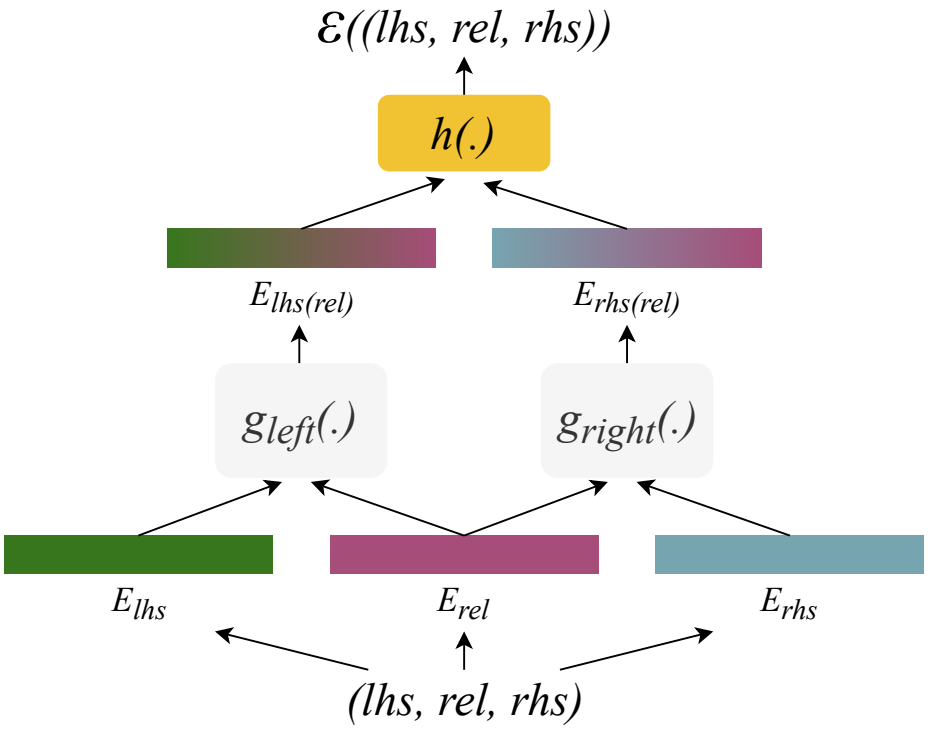

Figure 4.1: SME function: scheme of the neural network architecture. The words (lhs and rhs) and the relation type (rel) are mapped to their corresponding embeddings (E_{lhs}, E_{rhs} and E_{rel}). Then they are combined using functions g_{left} and g_{right}, resulting in the relation-dependent embeddings ($E_{lhs(rel)}$ and $E_{rhs(rel)}$). Finally, these new embeddings are *matched* using function h, to produce the value of the energy of the input triple.

5. Experiment

In this study we aim to evaluate the quality of the embeddings extracted from lexical networks obtained under different approaches. To do so, for each of the three major groups of methods for graph embedding we select one as a representative, as follows:

– for matrix factorisation (MF) we use the implementation and model introduced in [19];

– for random walk (RW) we use the system presented in [22];

– for edge reconstruction (ER) we adapt the implementation of the SME model from [23].

Moreover, we study two fundamentally different semantic networks, i.e. WordNet (WN) and Small World of Words (SWOW), and apply the MF, RW and ER methods to obtain word embeddings from each of the graphs. This results in six substantially different models: three for WordNet (denoted *MF WN, RW WN, ER WN*) and three for SWOW (denoted *MF SWOW, RW SWOW, ER SWOW*). We explore a number of variants within each experiment targeting a given method and type of graph. The experimental space is dependent on both elements and the details are presented in the respective subsections.

For a wider assessment, we also include in the experiment one of the best corpus-based embedding models, *GloVe* [2], in order to compare the overall performance of the embeddings extracted from the two different semantic proxies (as introduced in Section 3.1), that is text and graphs.

Additionally, as a baseline in the experiments, we include the evaluation of randomly initialised embeddings (denoted *Random*), as a model carrying no lexical semantics, but distributing the vectors in the space in a random way. To generate such model, we use the LeCun uniform initialisation [25], commonly used for the initialisation of the embedding layers in neural networks (also utilised by the SME method). The values are drawn from a uniform distribution within $[-limit, limit]$, where: $limit = \sqrt{\frac{3}{embedding\_dimension}}$.

5.1. Training of WordNet models

5.1.1. Matrix Factorisation on WordNet

The selected Matrix Factorisation method simulates an infinite random walk by computing the Katz index on the graph's adjacency matrix. This operation involves the inversion of the matrix, which is

computationally very expensive and thus, can be challenging for larger graphs. Since the full WordNet graph in version 3.0 contains almost 150000 words, it was necessary to restrict the vocabulary used for the experiments due to the existing computational resource limitations.

The vocabulary was selected in a process where first, in order to ensure a high coverage of the evaluation sets, the words occurring in the test sets were guaranteed to be retained. Subsequently, the remaining vocabulary was sorted descendingly by the amount of relations that each word was involved in (i.e. the number of incoming and outgoing edges from all the synsets the word belonged to). This was to ensure the highly-connected nodes in the graph are retained, in order to increase the linkage between the words, when a subgraph of WordNet is used. Ordered in such a way, the list of all words in the full graph is denoted as the *full vocabulary*.

In the experiments, Saedi et al. [19] explored the influence of the size of the vocabulary used to extract word embeddings from WordNet on their performance in the similarity task. They report on the results obtained using a vocabulary of 25, 30, 45 and 60 thousand words (denoted 25k, 30k, 45k, 60k, respectively). The 60k vocabulary is the largest tested dataset, due to the resource limitations. The results show that the performance of the embeddings consistently increases with the additional parts of WordNet being included. The key conclusion of that research was that the WordNet-based embeddings are highly competitive with the mainstream corpora-based models, achieving better results than the widely used *word2vec* model.

For the sake of comparability between the methods, the **60k vocabulary** is used as the base for the input of all the methods used to extract embeddings from lexical graphs.

5.1.2. Random Walk on WordNet

The random walk methodology presented in [22] was applied to the full WordNet 3.0 graph. Apart from the words encountered in the synsets, the *glosses* were also used. A gloss is a brief definition of a synset, with optional example sentences, allowing the users to quickly grasp the concept's meaning. This makes the method a hybrid, combining the lexical information encoded both in the semantic graph and in the text of the glosses. For the sake of a fair comparison with the remaining methods, which rely exclusively on the semantic graph, we adapted that model to use only the graph information, as well as only the restricted 60k vocabulary.

The publicly available implementation of the system[1] allows for generating a *synthetic corpus* based on two data files: a *dictionary* and a *knowledge base*.

The dictionary contains a list of words (vocabulary), each mapped to a list of nodes in the graph that it belongs to. There is also a possibility of assigning weights to each of the nodes, which can be understood as the frequency of a sense in which a given word occurs. This information is estimated for some of the words by the linguists constructing the WordNet graph.

[1] Available at `https://github.com/asoroa/ukb`

The knowledge base is an annotated edge list: each entry consists of the identifiers of the lhs and rhs nodes, as well as some optional information, such as the type, source and weight of the relation.

In order to adapt the system to align with the requirements of our study, we generated new data files, containing only the words and concepts from the restricted 60k vocabulary. The generation was implemented using the WordNet Corpus Reader (from the Natural Language Toolkit [26]), which provides a simple and efficient interface for accessing the WordNet data in an object-oriented manner. This allows for a straightforward synset lookup for a given word, enumeration of the related synsets or lemmas, etc.

The dictionary was generated based on the 60k vocabulary described in Section 5.1.1. For each word in the vocabulary, we retrieved all synsets that the word belongs to, and their identifiers were put in the dictionary entry describing the word. Subsequently, the dictionary was scanned for all the occurring synset identifiers (forming a *synset whitelist*), which were then used for the generation of the knowledge base file. In the file we have included all WordNet relation instances, such that both lhs and rhs synsets were on the whitelist.

With these data files prepared, we ran the random walk corpus generation script, provided by the authors of [22]. Following these authors, we also generate 70 million synthetic sentences based on the data files, amounting to 536.6 million tokens. On such a synthetic corpus, we trained a *word2vec* Skip-Gram model [15], using the Gensim library [27].

5.1.3. Edge Reconstruction on WordNet

The authors of [23] conduct experiments on WordNet. However, they use an independently filtered dataset of around 41k synsets. Since in our experiments we use word-based vocabulary, the dataset required for this method had to be generated from scratch.

We follow a similar procedure as described in 5.1.2. For each word (lhs) in the vocabulary, we find all synsets ($lhs_{synsets}$) that it belongs to. For each of the $lhs_{synsets}$, we find all related synsets $rhs_{synsets}$ (using the 22 relation types between synsets[2]) and for each of the $rhs_{synsets}$ we find all words rhs belonging to that synset. We generate all triples $< lhs, rel, rhs >$ such that both lhs and rhs are in the vocabulary and rel is a relation connecting lhs_{synset} and rhs_{synset} to which lhs and rhs belong to, respectively.

Apart from that, we find all words rhs related to lhs through lemma-based relations,[3] and again generate all triples $< lhs, rel, rhs >$ such that both lhs and rhs are in the vocabulary and rel is a lemma-based relation between them.

[2]WordNet defines the following relation types between synsets: *hypernym, instance hypernym, hyponym, instance hyponym, member holonym, substance holonym, part holonym, member meronym, substance meronym, part meronym, topic domain, in topic domain, region domain, in region domain, usage domain, in usage domain, attribute, entailment, cause, also see, verb group, similar to*.

[3]WordNet defines 3 lemma-based relations: *antonym, derivationally related form, pertainym*.

Synonymy relation is implicitly encoded in WordNet by words being grouped in synsets. Thus, we explicitly generate triples for the synonymy relation between all pairs of words in a synset, such that both words are in the vocabulary (i.e. only if there are two or more in-vocabulary words within a synset).

Influence of the subgraph size. We explore how the amount of used vocabulary items (and therefore, relations) influences the quality of the obtained word embeddings by creating datasets for various sizes of the vocabulary. We generate the lists of triples for 15k, 30k, 45k, 60k and 90k vocabularies. The results of the intrinsic evaluation of the vectors are presented in Section 6.1.1.

5.2. Training of SWOW models

The raw data collected by the authors of [8] was preprocessed and normalised (through fixing typos, americanising all word forms, removing participants with many unknown or missing responses, etc.). Additionally, the dataset is balanced by choosing exactly 100 participants' responses for each of the 12217 cues. Following the authors, we will denote this version of the dataset as SWOW-EN.

In [8], two variants of the graph are explored: one induced by relation $R1$ (where only the first response to the cue word by the human subject is taken into account and the remaining are discarded), and another induced by relation $R123$, where all three responses are aggregated, regardless of their position. In our evaluation, we always use all three associations, as De Deyne et al. [8] show that such models consistently perform better than the ones using only the first (strongest) association.

5.2.1. Matrix Factorisation on SWOW

In [8], the authors evaluate three measures of semantic similarity. First, the *associative strength*, as the simplest measure of semantic relatedness, defined as the probability of responding with the word w, given c as the cue. However, this simple measure is only capturing the *local* similarity, without taking into account the information stored in the full graph. Thus, the second measure explored is the *positive pointwise mutual information* (PPMI), computed for each *cue-response* pair using the following formula:

$$\begin{aligned} PPMI(r,c) &= \max\left(0, \log_2\left(\frac{p(r|c)}{p(r)}\right)\right) \\ &= \max\left(0, \log_2\left(\frac{p(r|c)}{\sum_{i=1}^{N} p(r|c_i)p(c)}\right)\right) \\ &= \max\left(0, \log_2\left(\frac{p(r|c)N}{\sum_{i=1}^{N} p(r|c_i)}\right)\right) \end{aligned} \tag{5.1}$$

where $p(r|c)$ denotes the probability of giving the response r to the cue word c and N is the number of cue words in the model.

PPMI extends the associative strength measure by considering the distributional information across the full graph, however still in a *local* way, since it considers all the responses with regard to a specific cue word. Therefore, the third measure considered by the authors is the Katz index, using spreading

activation mechanism in order to include the indirect paths between the nodes, thus capturing the global perspective in the network.

The detailed comparison of these three measures is presented in [8]. In this work, we use the third approach, as it proved to be the best overall in the evaluations presented by the authors. Following the authors' methodology, we use the existing implementation,[4] to extract the adjacency matrix, apply the PPMI transformation on it, and finally compute the infinite random walk by solving Equation 4.3. Each of these steps is additionally followed by L1 normalisation.

Subsequently, in order to obtain comparable models, we apply dimensionality reduction to the resulting matrix using Principal Components Analysis (PCA), to obtain 300-dimensional word embeddings.

5.2.2. Random Walk on SWOW

For SWOW, we follow a similar procedure as in the case of random walks on WordNet. As described in Section 5.1.2, in order to generate the synthetic corpus, we need *a dictionary* and *a knowledge base* for the SWOW-EN dataset. Since there is no notion of a synset in SWOW, we adapt the method to the word-based dataset by simply treating each word as a separate "synset" (node) containing a single lemma. Therefore, the knowledge base file consists of a list of $< cue, response >$ pairs, constructed easily based on the SWOW-EN data. The dictionary is a mapping of node identifiers to words belonging to the node. Since all the nodes contain a single lemma, the dictionary file is a list of $< word, word >$ pairs, where the former is the node identifier, and the latter the word form itself.[5]

Again, we run the corpus generation script based on these data files and generate 70 million synthetic contexts, accounting for 536.5 million tokens. Similarly as for the WordNet data, we train a word2vec Skip-Gram model to obtain word embeddings based on this corpus.

5.2.3. Edge Reconstruction on SWOW

For the triple-based data based on SWOW, we distinguish three types of relations: $R1$, $R2$, $R3$, corresponding to the first, second and third response to a given cue, respectively.

We adapt the authors' code to generate the association strengths for each of the relations separately. Through this, for each of the cue words, we get a list of words that were provided as the first, the second and the third association, separately. Based on this data, we can generate in a straightforward way the final list of triples for the SME method by joining each cue word with each of the first associations using the $R1$ relation, etc. The resulting dataset consists of close to 1.5 million triples.

[4] Available at: `https://github.com/SimonDeDeyne/SWOWEN-2018`

[5] The identifier could be represented e.g. in a numerical form, however since all the word forms are unique, they can simply serve the role of identifiers.

6. Evaluation

The mainstream way of assessing the quality of word embeddings is the, so-called, *intrinsic evaluation*. It is a direct evaluation of the obtained vectors in the tasks of semantic similarity and relatedness.

Another approach is the *extrinsic evaluation*, where the embeddings are evaluated indirectly through their usage in downstream tasks. The performance of complex systems that use pretrained embeddings for further processing of sentences, paragraphs, etc. is evaluated. This should provide insight into an important question of how the quality of direct encoding of lexical semantics (assessed in the intrinsic tasks) is reflected in the performance of complex systems in downstream tasks.

6.1. Intrinsic tasks

We evaluate the embeddings in semantic similarity and relatedness tasks, where the similarity of the vectors is matched against gold standard scores established by humans. Each dataset consists of a list of word pairs that have been ranked by human scorers, together with the score. The process of obtaining such ranks involves gathering the individual scores from multiple participants (e.g. through Amazon Mechanical Turk platform), normalisation and cross-validation of the scores, in order to assure high reliability of the final scores obtained for each word pair.

It is worth noting that even though the method of evaluation in both tasks is the same, the concepts of similarity and relatedness are not equivalent. Similarity is exemplified by synonymy: a pair of synonymous words is highly similar. Relatedness, on the other hand, can be seen as the level of association between two concepts. E.g. the words *coffee* and *cup* are not similar, but definitely related, while *cup* and *mug* are highly similar and related.

This distinction has not always been expressed explicitly to the participants of the study, which for example, led to dividing a state-of-the-art WordSimilarity-353 dataset [28] into two: WordSim353-Sim (for semantic similarity) and WordSim353-Rel (for conceptual relatedness) by Agirre et al. [29].

We distinguish these concepts and use 6 datasets in the intrinsic evaluation: 3 for semantic similarity and 3 for relatedness.

Semantic similarity datasets used in the evaluation are:

– **SimLex-999** [30], 999 word pairs, rated explicitly for semantic similarity;

– **RG1965** [31], 65 word pairs rated for semantic similarity;

– **WS353-Sim** [29], a subset of WordSimilarity-353 [28], 203 word pairs selected as rating similarity of the concepts.

Semantic relatedness datasets used in the evaluation are:

– **WS353-Rel** [29], another subset of WordSimilarity-353 [28], 252 word pairs selected as rating relatedness of the concepts;

– **MEN** [32], 3000 word pairs, crowdsourced through Amazon Mechanical Turk, annotated by participants with a binary choice of a more related pair out of two pairs displayed;

– **MTurk-771** [33], 771 word pairs rated for relatedness using Amazon Mechanical Turk platform.

The scores in the data files serve as the *gold standard*. In order to evaluate the embeddings, we need to compute such scores based on the respective vectors and compare them. A commonly used technique is computing a *cosine similarity* of the vectors (Equation 4.1).

Scores obtained in this way are matched against the gold standard using *Spearman rank-order correlation coefficient (ρ)*, which measures the strength and direction of the monotonic relationship between two ranked variables. Therefore, the gold standard and the model-based scores are first ranked, and the value of the metric is computed using the following formula:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{6.1}$$

where d_i is the difference in the ranks for item i. A Spearman correlation $\rho = 1$ indicates that the ranks are identical, $\rho = -1$ indicates that the ranks are exactly opposite, and $\rho = 0$ indicates no correlation between the ranks. In all plots and result tables we report the value of the correlation multiplied by 100.

6.1.1. Results of the WordNet models

Matrix Factorisation on WordNet. Saedi et al. [19] resort to embeddings with vector dimension $d = 850$. Since the last phase of obtaining these embeddings is the dimensionality reduction using PCA, the vector dimensions are sorted by descending variance and the first (most informative) n dimensions are retained as embeddings. Therefore, following this insight, we can retain only the first 300 dimensions of the existing embeddings in order to obtain a model that is comparable to other models in our experiments.

We plot the scores of both models (for $d = \{300, 850\}$, in light and dark red, respectively) in Figure 6.1 (among other models, described below). By comparing the two, we can confirm that shortening the vectors did not affect negatively the scores (the only drop of 0.2 point is present for the SimLex-999 testset). In fact, it resulted in slightly better scores in 4 testsets (RG1965, WS353-Sim, WS353-Rel and MTurk-771).

Random Walk on WordNet. We explore three variants of the RW method, differing in the data used for creating the synthetic corpus and retaining the same size of the corpus (70M sentences). We use the 60k vocabulary model for comparability. In addition, we evaluate the impact on the performance of

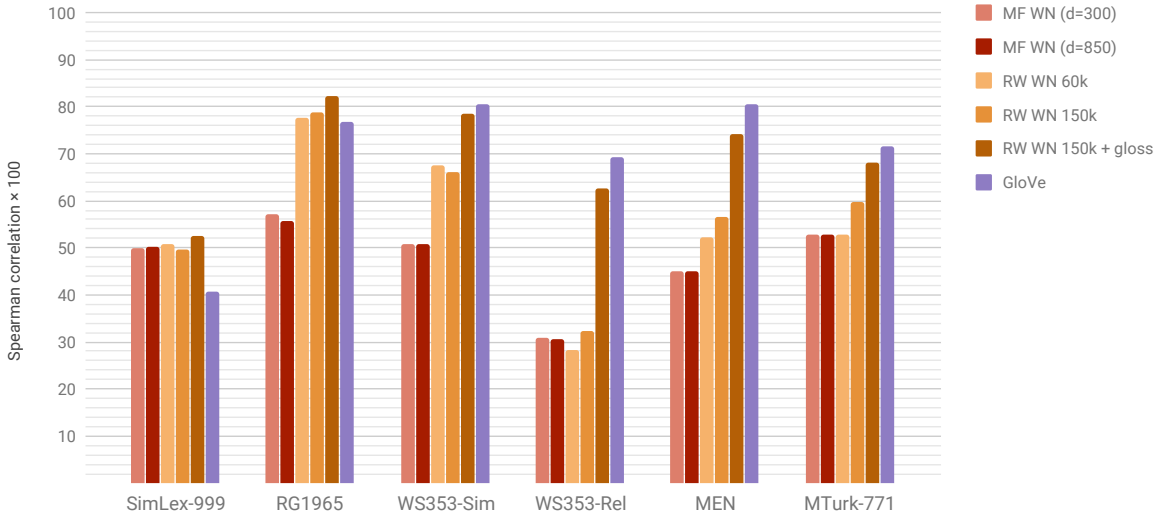

Figure 6.1: Plot of Table A.1 (Appendix A). Results of the intrinsic evaluation of the matrix factorisation (MF) models for two embedding dimensions: 300 (light red) and 850 (dark red); random walk (RW) models based on: 1) the 60k WordNet vocabulary (the lightest orange), 2) the full WordNet graph of almost 150k vocabulary (middle orange), 3) the full WordNet with gloss relations (dark orange); and the text-based model, GloVe (purple). Presented scores (vertical axis) are Spearman's rank-order correlation coefficients of the obtained vector similarities against the gold standard defined by each of the six testsets (horizontal axis).

using the full graph (the full vocabulary), as well as the original model from [22] that additionally uses the WordNet glosses.

The results are presented in Figure 6.1 (shades of orange). As expected, the model based on the full graph with glosses consistently outperforms the other two. What is worth noting, though, is that the superiority is much more visible in the relatedness task (WS353-Rel, MEN, MTurk-771), while not so much in the similarity task (SimLex-999, RG1965, WS353-Sim). This indicates that the gloss relations, which add textual information to the graph-based model, allow for bringing a significant amount of information about relatedness of the concepts, while do not improve on the more restrictive relation of similarity.

The model based on the smaller (60k) vocabulary performs closely to the one using the full WordNet graph (without glosses). There is a slight advantage of the former model in the similarity task, but the roles change in favour of the latter in the relatedness task. This superiority in assessing relatedness is probably caused by the lack of some significant connections in the restricted subgraph.

Comparison with a text-based model. The last model plotted in Figure 6.1 (in light purple) is GloVe [2], added as a representative for text-based embeddings. All WordNet-based methods outperform the text-based method on the hardest testset of similarity (SimLex-999), while the random walk based methods also outperform it in the RG1965 testset. In WS353-Sim and all testsets in relatedness task, the

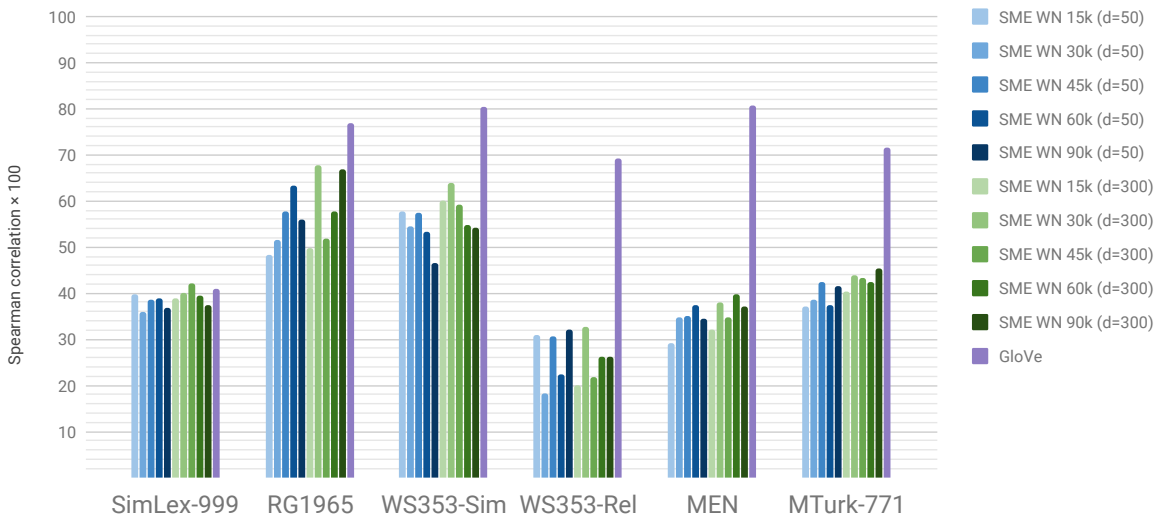

Figure 6.2: Plot of Table A.2 (Appendix A). Results of the intrinsic evaluation of the SME models for increasing size of the WordNet subgraph (15k, 30k, 45k, 60k, 90k) and two embedding dimensions: 50 (shades of blue) and 300 (shades of green), and the text-based model, GloVe (purple). Presented scores (vertical axis) are Spearman's rank-order correlation coefficients of the obtained vector similarities against the gold standard defined by each of the six testsets (horizontal axis).

text-based method outperforms the others, scoring several points more than the random walk method on the full graph with glosses.

This may suggest that the relatedness relation can be better captured on the basis of the co-occurrence of words, rather than using the inference-based network.

Edge Reconstruction on WordNet. Since this method was not extensively explored in the context of WordNet, we seek to gain insight on some of its characteristics by evaluating several models that result from varying two aspects. Similarly as in [19], we explore the influence of the size of the graph on the performance of the embeddings. Saedi et al. [19] show that using matrix factorisation on larger subgraphs consistently improves the performance of the embeddings. We train the models for the vocabularies of 15k, 30k, 45k, 60k and 90k words. Additionally we explore two embedding dimensions: $d = 50$ (shorter vectors) and $d = 300$ (longer vectors).

The results are shown in Figure 6.2. The models using shorter embeddings are presented in shades of blue, and the longer embeddings in shades of green (the darker the shade, the larger the WordNet subgraph). As in Figure 6.1, the GloVe model's performance is also plotted for reference (in light purple).

We expect a better performance from the larger vectors, as they are able to encode more information (knowledge) in the weights. As seen in Figure 6.2, the models using a larger dimension in general perform marginally better than their counterparts using shorter embeddings (i.e. when comparing the models with the same vocabulary, e.g. $(15k, d = 50)$ and $(15k, d = 300)$, $(30k, d = 50)$ and $(30k, d = 300)$, etc.). For example, in all testsets, the 30k vocabulary model performs better when using the larger embeddings;

similarly, all models follow this pattern for the MTurk-771 and WS353-Sim testsets. On the whole, the larger embeddings perform better in 22 out of 30 cases. The advantage is, though, not so substantial, taken into account the overhead of training the larger models (1-2 vs. 5-6 days of training). This might suggest that e.g. the shorter vectors are already able to encode quite well the information present in the datasets, or that the training of the longer vectors was performed not long enough.[1]

Regarding the other aspect, i.e. the size of the vocabulary, the desired behaviour would be an increased performance with an increased size of the subgraph used. This pattern, however, is not so visible. Partially it can be seen, e.g. in the results of both blue and green models on RG1965 or the green models on WS353-Rel, where 4 out of 5 models follow this pattern. In general, the largest 90k vocabulary models perform better than the smallest, 15k vocabulary (for both embedding dimensions, on 4 out of 6 testsets: RG1965, WS353-Rel, MEN, MTurk-771). Interestingly, for WS353-Sim the pattern seems actually reversed: the larger the vocabulary, the lower the scores obtained on this testset. In SimLex-999, MEN and MTurk-771, for each set of models, the scores seem rather flat, with the deltas between the minimal and maximal value in the range of 4-8 points.

This may result from the fact that (as detailed in Section 5.1.1) the test vocabulary was retained with priority, therefore all the direct relations between the test words are already contained in the smallest dataset (15k vocabulary). The edge reconstruction based methods focus usually on the 1st order proximity (the direct neighbours in the graph). If there is no explicitly expressed transitivity of the relations, adding the larger parts of graph to the dataset are not as beneficial for these models, as for the matrix factorisation or random walk based methods, that are able to exploit the indirect paths between the words.

Comparison with a text-based model. While on SimLex-999 the results are very competitive compared to the text-based model (Glove is outperformed by the 45k and 60k vocabulary models using larger embedding vectors), in all other testsets GloVe outperforms all edge reconstruction based models by a significant margin.

It is worth noting that these results should be taken with a grain of salt. Training of the SME models contains a random factor (the initialisation of the weights in the network). Training multiple models for each setting would allow for obtaining average and standard deviation of each model's performance, making the results more reliable. However, due to limitations on computational resources available, we were unable to perform such training in this part of the experiment (training of a single model would often take even a week). Multiple models were trained in selected experiments (as described below).

6.1.2. Results of the SWOW models

Matrix Factorisation on SWOW. De Deyne et al. [8] explore two variants of the graph: induced by relation *R1*, i.e. only the strongest association, and induced by relation *R123*, i.e. all three associations aggregated. We obtain the models using the available implementation[2], and then perform PCA for dimensionality reduction. We evaluate two embedding dimensions: $d = \{300, 850\}$, as in the WordNet-based

[1] Due to the limitations on computational resources, we were unable to explore this further.

[2] Available at: https://github.com/SimonDeDeyne/SWOWEN-2018

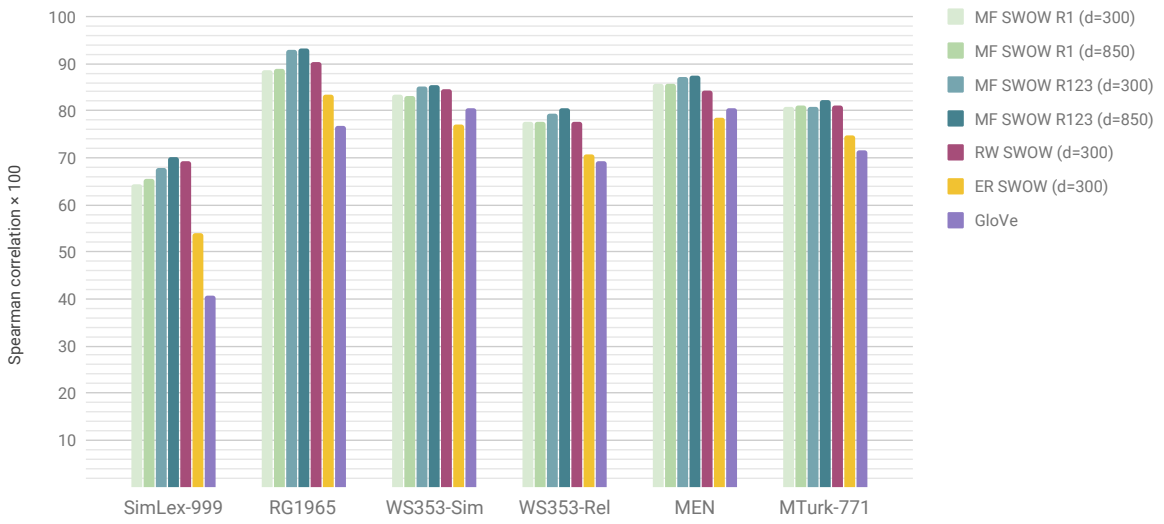

Figure 6.3: Plot of Table A.3 (Appendix A). Results of the intrinsic evaluation of the models based on the SWOW graph: 1) four models using matrix factorisation: two based on relation *R1* (shades of green) and two based on relation *R123* (shades of blue), using embedding dimensions of 300 (lighter shade) and 850 (darker shade); 2) random walk based model (magenta); 3) edge reconstruction based model (yellow); 4) text-based model, GloVe (purple). Presented scores (vertical axis) are Spearman's rank-order correlation coefficients of the obtained vector similarities against the gold standard defined by each of the six testsets (horizontal axis).

method. The results are presented in Figure 6.3 (two models using *R1* in light green and two models using *R123* in blue).

The models using larger embeddings are in general performing slightly better than the respective models using shorter vectors, with deltas ranging from 0.1 to 1.2 points for the *R1* models, and from 0.3 to 2.2 points for the *R123* models. The superiority of the *R123* over the *R1* models, advocated in [8], is still present after the dimensionality reduction. It suggests that the first (thus, strongest) association is highly informative in the similarity and relatedness tasks, but the other, weaker associations are further enriching the encoded information [8].

Random Walk on SWOW. We evaluate one setting of the random walk for SWOW, using the full graph and all three associations. As described in Section 5.2.3, we generate a corpus of 70M synthetic sentences based on the graph and train a Skip-Gram model over it. We use the same set of parameters as [22]: 3 epochs, 5 negative samples, context window of size 5, embedding dimension of 300, and all other parameters with default values as provided by the Gensim library [27].

We train three models, initialised with different random seeds and report the average of the scores. The standard deviation was low, ranging from 0.1 to 0.5 point. The averaged results of the intrinsic evaluation are presented in Figure 6.3 (in magenta). The scores are consistently lower than the best model based on matrix factorisation (with larger embeddings, $d = 850$), though very competitive. When

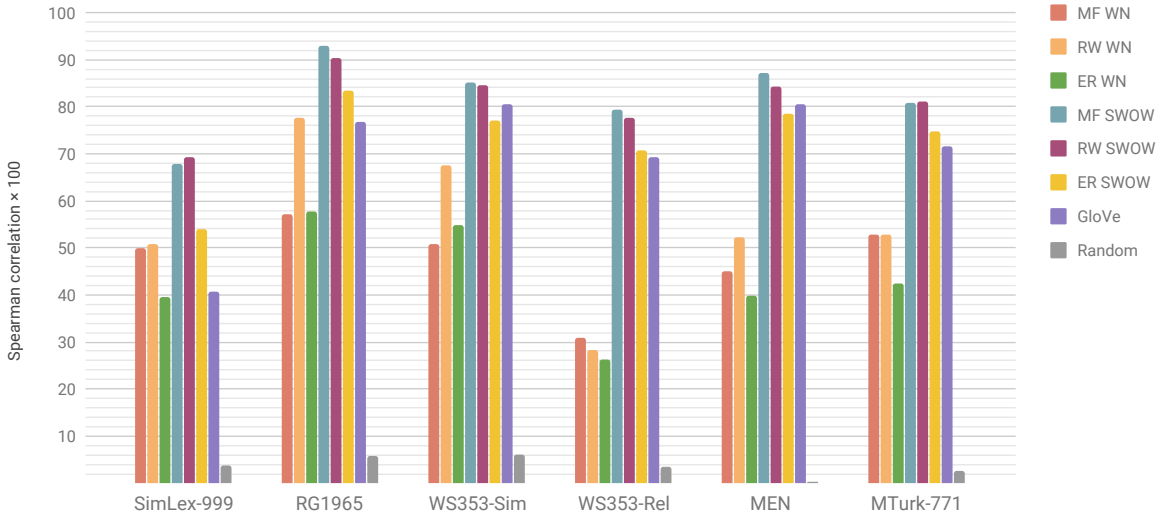

Figure 6.4: Plot of Table A.4 (Appendix A). Results of the intrinsic evaluation of the 8 models for comparison. All WordNet models are based on the same vocabulary subset (60k). The embedding dimension is 300. The *Random* baseline model is plotted in grey.

comparing the models with the same embedding dimension, the RW model already outperforms the MF model in two testsets (SimLex-999 and MTurk-771) and scores closely in the remaining ones.

This similarity in performance of these two methods can originate from the underlying similarity of the methods: both explore the graph in a sort of a random walk, which allows them to encode similar information about the graph structure.

Edge Reconstruction on SWOW. We train the model as described in Section 5.2.3, using the dataset based on all three associations. Again, we train 3 models and report the average results in Figure 6.3 (in yellow). The standard deviation was higher than in the RW model, ranging from 3.7 to 6.2 points.

The edge reconstruction based model is consistently performing worse than the other models based on SWOW, with deltas ranging from 3 to 7 points (compared to the lowest score among the four MF models and the RW model). This suggests that incorporating more than just the local neighbourhood information is crucial in encoding lexical semantics.

Comparison with a text-based model. All matrix factorisation based models and the random walk model outperform GloVe by a significant margin. The edge reconstruction based model is competitive with GloVe, outperforming it in 4 out of 6 testsets, and scoring closely in the remaining two. This shows that the association-based models like SWOW, are a very strong indication of similarity and relatedness, yielding significantly better results than a mainstream co-occurrence based model like GloVe, trained on a large textual corpus of 840B tokens.

6.1.3. Final results of intrinsic evaluation

For the final comparison, for each method-graph setting we choose one representative model out of the variants presented above. As comparable settings, we choose the largest common vocabulary for WordNet (60k), SWOW dataset based on all three associations (*R123*), and an embedding dimension of 300. We train 3 models for each random walk based and edge reconstruction based method and report the averaged results.

In Figure 6.4 we report the results for three WordNet-based models (*MF WN*, *RW WN* and *ER WN*, using matrix factorisation, random walk and edge reconstruction, respectively), three SWOW-based models (*MF SWOW*, *RW SWOW* and *ER SWOW*), one text-based model (*GloVe*) and a *Random* baseline model (as described in the introduction of Chapter 5).

When comparing the lexical graph-based models, a clear advantage of the SWOW graph is visible: in each of the testsets, all three SWOW models outperform even the best model based on WordNet (*RW WN*). This suggests that the *feature-based model using free associations is better able to encode lexical semantics* in terms of similarity and relatedness of words.

The SWOW-based models also show a larger advantage over the WordNet-based models in the relatedness tasks, compared to the similsrity tasks. This, however, could be potentially of use for the applications where the distinction between similarity and relatedness is important. Several such applications are listed by Hill et al. [30], e.g. automatic generation of dictionaries, thesauri, ontologies and language correction tools, or machine translation systems. In such applications, the usage of WordNet might actually be advantageous, thanks to its more rigorous structure.

With regard to the methods, the edge reconstruction performs the worst (based on SWOW in all testsets, while on WordNet in 4 out of 6 testsets). *Matrix factorisation and random walk share the best results*: the former performs the best based on SWOW, while the latter on WordNet.

The reason for this may lie in the extent to which the graph is structured. Since matrix factorisation systematically and thoroughly covers the paths within the graph, it may alleviate the lack of a formal structure of the semantic information encoded in an association-based graph. On the other hand, the random walk methods are known to provide sub-optimal sampling by being biased towards the nodes with many edges. This effect might be softened by the systematic structure and hierarchy present in the WordNet graph.

6.2. Extrinsic tasks

In order to evaluate the embeddings in the extrinsic tasks, we need two elements: a) the datasets suitable for the downstream tasks, and b) the model that will be solving those tasks, that will be using the word embeddings, and whose performance will be directly evaluated, thus giving an indirect indication of the word embedding impact and quality.

Hence, our framework for extrinsic evaluation is based on two recent works in the area of natural language processing and sentence understanding. The first one, the GLUE Benchmark [3], provides

several datasets for various sentence understanding tasks. The second one, the JIANT Framework [34], is an implementation of a framework allowing for building complex models for sentence understanding, as well as for training and evaluating such models.

6.2.1. GLUE Benchmark

The GLUE Benchmark [3] is a collection of resources for training, evaluating, and analysing natural language understanding systems. It defines 9 tasks based on existing, established datasets, relying on sentence or sentence-pair understanding. The tasks were selected so that they cover various domains, different data sizes and difficulty levels. The goal of the benchmark is to promote multitask and transfer learning in the natural language understanding systems.

Each dataset is divided into three disjoint sets for training, validation (development) and evaluation (test). In order to make the competition not trivial, the test sets are unlabelled: the labels are held private. A model can be evaluated by uploading the predictions obtained for all tasks to the project website[3], where it gets automatically scored. However, the process of uploading and evaluating is restricted: at least a preprint of a publication is required and a limit of two submissions per day is imposed. Therefore, our evaluation is based on the results obtained on the validation (development) set.

Due to the limitations on computational resources available, we selected 5 tasks out of the 9 included in GLUE, for which a multiple model training is feasible for all the embedding models. We describe the tasks shortly here, while the detailed descriptions can be accessed through the project web page.[4] As noted below, several samples from the datasets are presented in Table 6.1.

CoLA. Corpus of Linguistic Acceptability [35] is a set of over 9000 sentences annotated by experts for grammatical correctness. It is a binary classification task (assessing whether a given sentence is correct or not) with unbalanced classes. Therefore, Matthews Correlation Coefficient (MCC) [36] is used as the evaluation metric. The metric ranges from -1 to 1, where 0 stands for the performance of uninformed guessing.

SST-2. Stanford Sentiment Treebank [37] is a sentiment analysis task. It provides a collection of sentences from movie reviews, annotated by humans (through Amazon Mechanical Turk platform) with the sentiment of the sentence. The dataset utilised by GLUE uses the two-way class split (with only *positive* or *negative* class), as well as only sentence-level annotations, and consists of almost 70000 sentences. The evaluation metric on this dataset is accuracy.

MRPC. Microsoft Research Paraphrase Corpus [38] is a sentence-pair classification task, where the goal is to decide whether two given sentences are semantically equivalent (i.e. a paraphrase of each other). The corpus was automatically extracted from the news sources and annotated by humans for equivalence, and consists of almost 5500 sentence pairs. The evaluation metric on this dataset is accuracy and F1 score due to class imbalance.

[3] `www.gluebenchmark.com`
[4] `https://gluebenchmark.com/tasks`

RTE. Recognising Textual Entailment is a Natural Language Inference (NLI) task. In such tasks, given two sentences (a *premise* and a *hypothesis*), the goal is to decide whether the second (hypothesis): a) is *entailed* by the premise, or b) *contradicts* the premise, or c) is *neutral* with regard to the premise. This dataset, provided by GLUE, is a collection of RTE1 [39], RTE2 [40], RTE3 [41] and RTE5 [42], which were collected during a series of annual challenges for textual entailment. The corpus is based on news and Wikipedia text and contains over 5500 sentence pairs. Some of the datasets were prepared as two-class splits (*entailment/not_entailment*), while some as three-class splits (*entailment/contradiction/neutral*). For consistency, all three-class split datasets are converted to two-class splits by collapsing classes *neutral* and *contradiction* into a single *not_entailment* class. The evaluation metric of this dataset is accuracy.

WNLI. Winograd NLI is another Natural Language Inference corpus. It was generated by the authors of GLUE based on the Winograd Schema Challenge [43]. In this challenge, given a sentence, the system is supposed to replace a marked pronoun with one of the responses from a given list of choices. This task is converted by the authors of GLUE to a NLI task: the ambiguous pronoun is replaced by each possible referent (from the list of choices) and a sentence pair is generated for each of these as $< original\_sentence, substituted\_pronoun\_sentence >$. Thus, in the WNLI task, the goal is to predict the entailment relation between the $original\_sentence$ and the $substituted\_pronoun\_sentence$. The dataset consists of 780 sentence pairs and the metric used for evaluation is accuracy.

Samples from the datasets are presented in Table 6.1. For each task, two examples are included, one from a "positive" and one from a "negative" class (*correct/incorrect sentence* for CoLA, *positive/negative sentiment* for SST-2, *paraphrase/not paraphrase* for MRPC and *entailment/not entailment* for RTE and WNLI). It is worth noting that CoLA and SST-2 are single-sentence classification tasks, while the remaining three tasks receive as input two sentences, denoted in the table as **S1** and **S2** (for the MRPC task), or **P** (premise) and **H** (*hypothesis*) for the NLI tasks (RTE and WNLI).

6.2.2. JIANT Framework

The JIANT Framework [34] is a more flexible extension of the training and evaluation framework prepared by the authors of GLUE. It is a configuration-driven software toolkit for research on general-purpose natural language understanding systems. It was designed to facilitate work on multitask and transfer learning in tasks requiring sentence understanding.

The framework's architecture (Figure 6.5), presented in [1], is based on three layers:

1. input layer (based on character and/or word embeddings);
2. sentence encoder layer;
3. task-specific classifiers layer.

The sentence encoder is shared across all tasks, which allows for multitask and transfer learning. It receives as input the sequences of character and/or word embeddings (of a single or a pair of sentences),

Task	Input	*Output*
CoLA	If Ron knows whether to wear a tuxedo, and Caspar knows whether not to, do they know different things?	*Correct*
	David is a great artist, and when he does, his eyes squint at you.	*Incorrect*
SST-2	You don't have to know about music to appreciate the film's easygoing blend of comedy and romance.	*Positive*
	All that's missing is the spontaneity, originality and delight.	*Negative*
MRPC	**S1:** Fires in Spain's Extremadura region, which borders Portugal, have forced hundreds of people to evacuate their homes. **S2:** Fires in Spain 's Extremadura region bordering Portugal, and Avila province forced hundreds of people to leave their homes.	*Paraphrase*
	S1: Five more human cases of West Nile virus, were reported by the Mesa County Health Department on Wednesday. **S2:** As of this week, 103 human West Nile cases in 45 counties had been reported to the health department.	*Not paraphrase*
RTE	**P:** Yoko Ono, widow of murdered Beatles star John Lennon, has plastered the small German town of Langenhagen with backsides. **H:** Yoko Ono was John Lennon's wife.	*Entailment*
	P: On Feb. 1, 1945, the Polish government made Warsaw its capital, and an office for urban reconstruction was set up. **H:** Warsaw remained Poland's capital after the war.	*Not entailment*
WNLI	**P:** The dog chased the cat, which ran up a tree. It waited at the bottom. **H:** The dog waited at the bottom.	*Entailment*
	P: As Ollie carried Tommy up the long winding steps, his legs ached. **H:** Tommy's legs ached.	*Not entailment*

Table 6.1: Samples from the datasets of the selected GLUE tasks. *Note:* CoLA and SST-2 are single-sentence classification tasks, while the remaining tasks (MRPC, RTE and WNLI) receive as input a pair of sentences: S1 and S2 denote the first and the second sentence, P denotes a *premise* and H a *hypothesis*.

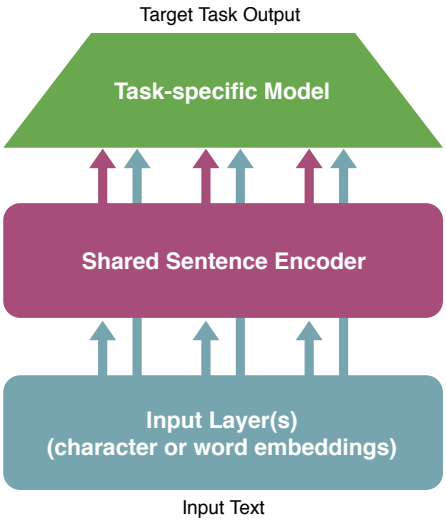

Figure 6.5: Three-layer architecture of the JIANT Framework. The figure is adapted from [1].

and its task is to produce a sentence encoding for each of the input sequences. These sentence embeddings are further fed to the task-specific classifiers, along with the original input layer embeddings. Based on this information, the classifiers perform predictions.

The common flow of training and evaluation of a model using JIANT consists of three phases: *pretraining*, *training* and *evaluation*. During pretraining both the shared sentence encoder and the task-specific classifiers are trained (using the *pretraining tasks*). Subsequently, the sentence encoder is frozen and in the training phase only the third layer (of the task-specific classifiers) is trained, using the *training tasks*. Finally, the evaluation is performed by using such full model for making predictions on the *test sets of the evaluation tasks*.

6.2.3. Training setup

The configuration of the task-specific classifiers for the GLUE tasks (and several other tasks defined and shared by the contributors) is provided by the authors of the framework.[5] Every configuration, however, is open for modifications as desired. For our experiments, we use the default parameters of the task-specific classifiers.

The primary purpose of the JIANT Framework is the evaluation of the second layer of the system, i.e. the various types of the sentence encoders. However, we seek to evaluate the quality of the *first* layer of the system, i.e. the word embedding layer. Therefore, we design the experiment in such a way that we *omit* the sentence-level embeddings. In fact, we use a predefined type of a void sentence encoder that receives the input and returns an empty output. Nonetheless, the first-layer embeddings are connected through a, so-called, skip-connection to the third layer (as they "skip over" the second layer). As a result,

[5] Available at: `https://github.com/nyu-mll/jiant`

the only input fed to the third layer are the sequences of word embeddings, on the basis of which the classifiers are trained and evaluated.

Since the sentence encoder is void, the pretraining phase is omitted. Thus, the flow in our experiment consists of two phases: training and evaluation. During training, only the third layer is trained (as the void sentence encoder has no trainable parameters). We train and evaluate the model on each of the 5 selected GLUE tasks, described in Section 6.2.1.

We use the various models of pretrained word embeddings (described in Section 6.1) as the input layer and the identical configuration of the third layer for each setup. Hence, the results of such evaluation are based solely on the information encoded in the word embeddings and in the third-level classifiers.

6.3. Results of the extrinsic evaluation

We evaluate the eight final models, as in Section 6.1.3, i.e. three models based on WordNet, three models based on SWOW, GloVe as a representative text-based model, and finally the baseline *Random* model consisting of the vectors initialised randomly, as described in Chapter 5. We run three experiments for each of the models, using different random seeds, and report the averaged results. Taking the same approach as the authors of GLUE, we normalise all scores to the range of $[0, 100]$ by computing them as follows:

– CoLA: Matthews Correlation Coefficient multiplied by 100;

– SST-2, RTE, WNLI: accuracy (in percent);

– MRPC: average of the two metrics, i.e. F1 score and accuracy (in percent).

Also, following the authors of GLUE, we present the unweighted average of the scores across all tasks (*Average score*). The final results are presented in Figure 6.6.

The first, general observation is that the advantage that the graph-based models exhibited in the intrinsic evaluation against the text-based model, are not so apparent in the results of the extrinsic evaluation. In fact, GloVe outperforms all the remaining models in 3 out of 5 tasks (SST-2, RTE and WNLI) and performs close to the top performing model on the remaining two tasks.

Regarding the two types of graphs, the clear relative superiority of the SWOW-based models over the WordNet-based ones is thoroughly mitigated in the extrinsic tasks. Similarly, the patterns regarding the performance of the different methods disappeared. Instead, the results of the graph-based models flattened, without distinctive top performing model. The large deltas between the highest and lowest scores, visible in the intrinsic evaluation, nearly vanished in the evaluation of downstream tasks. It is unclear whether the reason for this may lie in any characteristics of the tasks. It seems, however, that the diversity of the lexical semantic information encoded in the various models of embeddings, notably different at the low level, do not project the same type of influence on the results of the extrinsic evaluation.

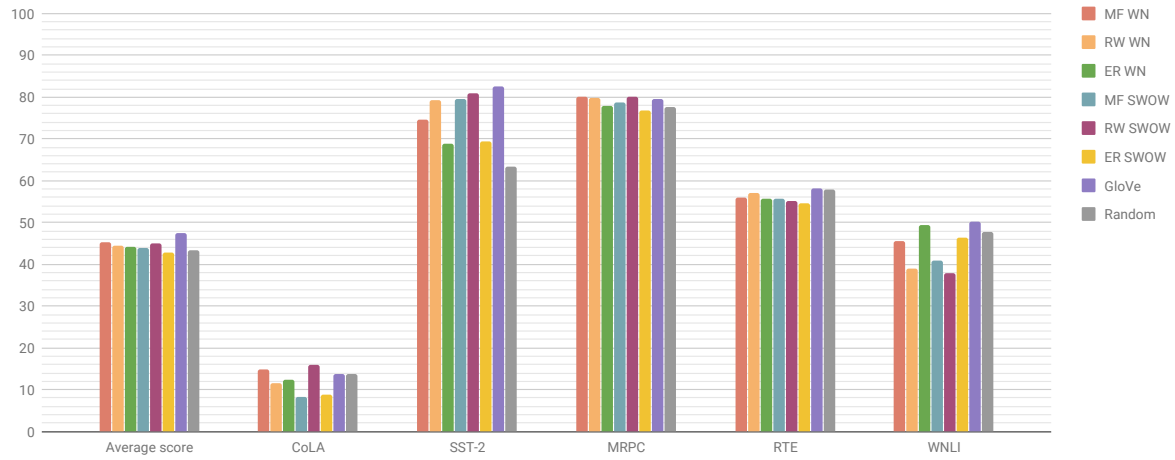

Figure 6.6: Plot of Table B.1 (Appendix B). Results of the extrinsic evaluation on selected GLUE tasks of the 8 models using different types of pretrained embeddings as input. The details about the metrics used for scoring are presented in the text. The colours are consistent with Figure 6.4.

This leads to another, crucial observation, which is the performance of the *Random* model. While in the intrinsic evaluation the random embeddings do not exhibit any potential in encoding lexical semantic information, in the extrinsic evaluation the model performs very competitively against the remaining models. It is among the top performing models on RTE (second best score), WNLI (third best score) and CoLA (third best score along with GloVe). This suggests that the intermediate representation of the sentences may be insensitive to the lexical information encoded solely in the pretrained word embeddings.

6.4. Diagnostic dataset

The authors of GLUE introduce an additional dataset together with the benchmark. It is a small, manually-curated testset that allows for the performance analysis of the language understanding systems. It is built for a fine-grained evaluation of the systems on a broad range of predefined linguistic phenomena in four major categories: Lexical Semantics, Predicate-Argument Structure, Logic and Knowledge.

The category of Lexical Semantics focuses on the issues of *word meaning*, from morphological negation (e.g. *agree - disagree*) to quantifiers (e.g. *most, some, all*). Predicate-Argument Structure concentrates on the issues of understanding how parts of the sentence are composed into the whole. This includes, handling e.g. prepositional phrases or relative clauses. The Logic category focuses on the ability to understand the semantics using the logical operators, such as negation, double negation, conjunction, disjunction, etc. The Knowledge category allows to evaluate whether the system not only correctly distinguishes the entailment relation, but also grounds the classification in some common sense and world knowledge, e.g. *There are amazing hikes around Mt. Fuji* entails *There are amazing hikes in Japan* but does not entail *There are amazing hikes in Nepal*.

Coarse-grained Categories	Fine-Grained Categories
Lexical Semantics	Lexical Entailment, Morphological Negation, Factivity, Symmetry/Collectivity, Redundancy, Named Entities, Quantifiers
Predicate-Argument Structure	Core Arguments, Prepositional Phrases, Ellipsis/Implicits, Anaphora/Coreference Active/Passive, Nominalization, Genitives/Partitives, Datives, Relative Clauses, Coordination Scope, Intersectivity, Restrictivity
Logic	Negation, Double Negation, Intervals/Numbers, Conjunction, Disjunction, Conditionals, Universal, Existential, Temporal, Upward Monotone, Downward Monotone, Non-Monotone
Knowledge	Common Sense, World Knowledge

Table 6.2: The fine-grained types of linguistic phenomena annotated in the diagnostic dataset (Section 6.4), organised under four major categories. *Note.* Reprinted from Wang et al., *GLUE: A multi-task benchmark and analysis platform for natural language understanding.*, 2019 [3]. The detailed description of each phenomenon can be found in [3] (Appendix E).

The list of all subcategories is presented in Table 6.2 and a detailed description of each can be found in [3] (Appendix E).

The dataset is a natural language inference (NLI) task using the three-class-split (*entailment, contradiction, neutral*). Each of the test samples is annotated with a set of linguistic phenomena that are involved in justifying the class label of the sample. Example tagged sentence-pairs are presented in Table 6.3.

Since the examples in the diagnostic set are hand-picked in order to expose certain linguistic phenomena, their distribution does not reflect the real distribution of the language in general. The testset is provided not as a benchmark (as the GLUE tasks), but as an analysis tool for qualitative model comparison and error analysis. Therefore, the performance scores should not be compared between the different categories for a given model, but various models should be compared within the same category.

6.4.1. Discussion of the results

Due to the class imbalance in the dataset, the R_3 coefficient (a three-class generalisation of the Matthews correlation coefficient [44]) is used for evaluation. Following the authors of GLUE, we report the scores multiplied by 100 (denoted $100R_3$), which puts them in the range of $[-100, 100]$, where 100 denotes a perfect correlation, -100 denotes a perfectly negative correlation, and 0 denotes no correlation between the variables.

Tags	**Premise**	**Hypothesis**	**Label**
Lexical entailment, Conditionals	The longer he stays in power, the harder it will be to exit.	The shorter he stays in power, the easier it will be to exit.	E
Active/Passive, Prepositional phrases	Soft plant parts and insects are eaten.	Cape sparrows eat seeds, along with soft plant parts and insects.	N
Relative clauses	The profits of the businesses that focused on branding were still negative.	The businesses that focused on branding still had negative profits.	E
Anaphora/Coreference, Double negation	A rabbi is at this wedding, standing right there standing behind that tree.	It's not the case that there is no rabbi at this wedding; he is right there standing behind that tree.	E
Genitives/Partitives, Negation	The Cape sparrow's population has not decreased significantly, and is not seriously threatened by human activities.	The population of the Cape sparrow has decreased significantly, and is seriously threatened by human activities.	C
Intersectivity, Downward monotone, Conditionals	You know that some life changing actions must be taken when grandma reacts with the sad emoji.	You know that some life-changing actions must be taken when grandma reacts with emoji.	N
Redundancy, Ellipsis/Implicits, World knowledge	David Tennant is the best Doctor in the series.	David Tennant is the best Doctor in the Doctor Who series.	E
Quantifiers, Universal	Everyone has a set of principles to live by.	No one has a set of principles to live by.	C
Quantifiers, Existential	Susan knows how turtles reproduce.	Someone knows how turtles reproduce.	E
Conjunction	Temperature must be just right.	Temperature and snow consistency must be just right.	N

Table 6.3: Examples from the diagnostic set, tagged with the phenomena they demonstrate. Each phenomenon belongs to one of four broad categories (see Table 6.2). Labels are *entailment* (E), *contradiction* (C) or *neutral* (N).

In general, the distributions of the scores among various models demonstrate a considerable level of similarity, with some phenomena being captured better or worse by certain models. The full evaluation results are presented in Appendix C. Here, we will focus on several scores considered interesting and potentially providing a valuable insight. The scores for selected subcategories and each whole category are presented in Figure 6.7. We will evaluate the models by relating their score to the one obtained by the baseline *Random* model.

In overview, there are several subcategories in which the models using pretrained embeddings improve the performance of the baseline *Random* model. A significant improvement is observed in Lexical Entailment (from the Lexical Semantics category), Active/Passive (covering the relationship between the active and passive voice) and Relative Clauses from the Predicate-Argument Structure category, as well as four subcategories of the Logic category: Universal and Existential (with significant improvement), Negation and Conditionals (with an improvement of the low negative score to a better score, however still negative).

In a number of categories, all but one model achieve better results than the baseline. These include: Quantifiers from the Lexical Semantics category, Prepositional Phrases, Genitives/Partitives and Ellipsis/Implicits from the Predicate-Argument Structure category, Conjunction from the Logic category, and World Knowledge from the Knowledge category.

This indicates that the pretrained embeddings can improve the system's ability to understand various lexical phenomena.

Interestingly, there are a few subcategories, in which the baseline model actually outperforms all remaining models. These are Anaphora/Coreference and Intersectivity from the Predicate-Argument Structure category, as well as Double Negation from the Logic category. This may be caused by the fact, that these phenomena are too complex to model using only a sequence of word embeddings and a require more expressive systems to capture the relationship between the words.

Finally, an important outcome of this evaluation is that in each of the four major categories, without the distinction of the subcategories (denoted *Whole category*), all models improve the scores of the baseline *Random* model.[6] This insight may indicate some advantage of using the pretrained embeddings in recognising certain low-level lexical phenomena in sentence understanding.

These results, however, should be taken with a grain of salt. The diagnostic dataset is a new tool for evaluation of the language understanding systems and has not yet been explored in the literature. It is unclear to what extent the results can be relied upon. Another important note is that the results have been gathered based on a single run of the evaluation. In order to obtain more credible results, it would be beneficial to conduct such evaluation multiple times and average the scores, which was impossible in the current study due to the limited computational resources available. Moreover, some linguistic insight into specific examples tagged with certain categories could prove valuable in further analysis of the results obtained using this dataset.

[6] With one exception of the RW WN model in the Knowledge category, where the score is lower by 0.5 point (compared to the baseline).

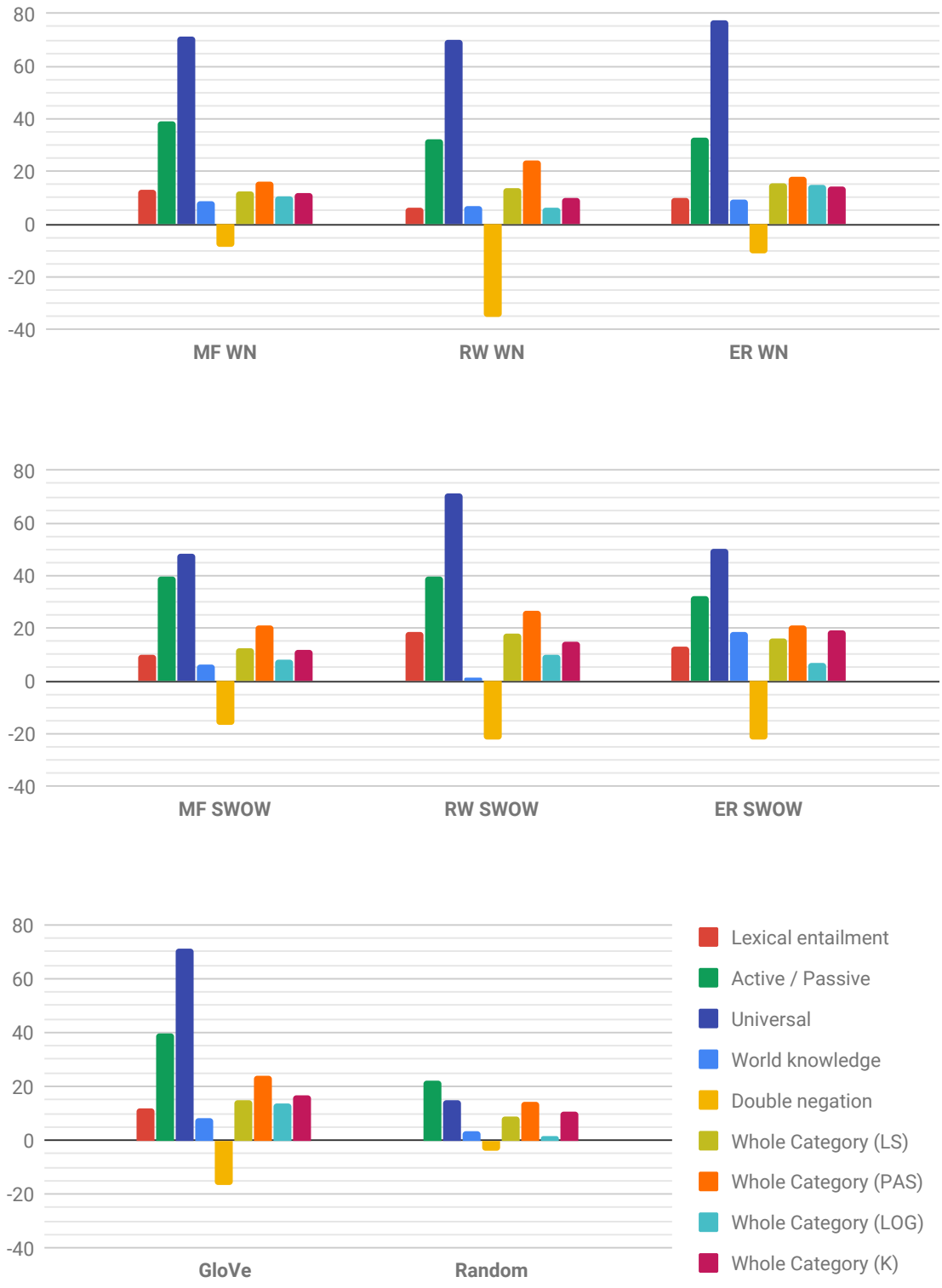

Figure 6.7: Scores for selected subcategories and each whole category of the Diagnostic dataset for all 8 models. The categories are: Lexical Semantics (LS), Predicate-Argument Structure (PAS), Logic (LOG) and Knowledge (K). The full evaluation results for all subcategories are presented in Appendix C.

7. Conclusion

In the present study, we explored the word embedding models based on different sources: textual corpora and two types of lexical graphs. The graphs encode the semantic information in a substantially different manner: in a structured hierarchy of concepts and using various lexical relations among the words (WordNet), and as a free-association network (Small World of Words). Moreover, we explore three different methods of obtaining the embeddings from the graphs: based on matrix factorisation, random walk and edge reconstruction.

The intrinsic evaluation, aiming to measure the model's ability to assess the semantic similarity and relatedness of the words based on the distribution of their embeddings, revealed a wide spectrum of the performance scores of the trained models. The results indicate that the embedding models based on lexical graphs are clearly competitive against the mainstream text-based model, with the best scoring graph-based model consistently outperforming GloVe by a substantial margin.

The best performing models are based on the Small World of Words graph - the feature-based model. This type of graphs can be built at a relatively affordable cost, as the data can be collected from lexical associations elicited from laypersons, as opposed to WordNet (an inference-based model), whose construction requires expert knowledge. This is of importance, as the current version of the English SWOW supports a vocabulary of only 12 thousand words, which can be very limiting when aiming to use the model in a language understanding system. Moreover, it is important for the support of new languages, other than already available in the SWOW project.

We would expect that the better performance of the model in the semantic similarity and relatedness tasks should improve the performance of the system resolving downstream tasks, e.g. based on sentence or sentence-pair classification. This, however, is not apparent in the obtained results. The diversity in the scores obtained in the intrinsic evaluation practically vanishes when the different models are used in the downstream tasks. Surprisingly, even the baseline model, consisting of the embeddings randomly spread in the embedding space, performs comparably to all the remaining models, that were informed by either large corpora of text or by the lexical graphs.

These observations raise interesting questions regarding the *universal* semantic information encoded in word embeddings, as well as their role in improving performance of the complex language understanding systems resolving downstream tasks. A better performance of the model in the intrinsic evaluation not only does not seem to guarantee an improvement in the downstream tasks, but seems not to affect it in a predictable way.

This appears in line with the research of Conneau et al. [45], concerning sentence embeddings. The authors aim to measure the correlation of the performance in the probing tasks (regarding various lexical aspects of the sentence) and the downstream tasks (such as the ones used in the current study, i.e. sentiment analysis, paraphrase detection, etc.). Their results suggests that there is little to co correlation between these scores, i.e. the model may encode various lexical phenomena very well, but at the same time perform poorly in the downstream tasks; as well as the other way around: the model may perform well in the downstream tasks even if it seems not to encode the given lexical phenomena.

7.1. Future work

The present study focused solely on word embeddings. The system used for the extrinsic evaluation was, in fact, rather basic, when compared to the current state-of-the-art language understanding systems. This was a conscious choice, as the study was targeting the low-level encoding of the lexical information extracted from graphs. However, it would be beneficial to evaluate the more expressive models, making use of multi-task and transfer learning, as well as cross-task knowledge sharing. Such models are highly encouraged and supported by the new frameworks, such as JIANT [34], which aim to facilitate and drive the research of language understanding systems.

Appendix A. Complete results of the intrinsic evaluation

	Similarity			Relatedness		
	Simlex-999	**RG1965**	**WS353-Sim**	**WS353-Rel**	**MEN**	**MTurk-771**
MF WN (d=300)	49.9	57.0	50.8	30.9	45.0	52.8
MF WN (d=850)	50.1	55.8	50.7	30.6	45.0	52.7
RW WN 60k	50.9	77.5	67.4	28.4	52.2	52.9
RW WN 150k	49.6	78.7	66.2	32.4	56.6	59.7
RW WN 150k + gloss	52.5	82.3	78.5	62.7	74.3	68.1
GloVe	40.8	76.9	80.4	69.3	80.6	71.6

Table A.1: Results of the intrinsic evaluation of the matrix factorisation (MF) models for two embedding dimensions: 300 and 850; random walk (RW) models based on different WordNet vocabularies (60k, 150k), with the usage of glosses where marked (+gloss); and the text-based model, GloVe. Presented scores (rows) are Spearman's rank-order correlation coefficients of the obtained vector similarities against the gold standard defined by each of the six testsets (columns).

	Similarity			Relatedness		
	Simlex-999	**RG1965**	**WS353-Sim**	**WS353-Rel**	**MEN**	**MTurk-771**
SME WN 15k (d=50)	39.9	48.4	57.8	30.9	29.3	37.2
SME WN 30k (d=50)	35.9	51.6	54.6	18.3	34.7	38.6
SME WN 45k (d=50)	38.7	57.8	57.4	30.7	35.0	42.4
SME WN 60k (d=50)	38.8	63.2	53.2	22.4	37.5	37.3
SME WN 90k (d=50)	36.9	56.1	46.4	32.1	34.4	41.6
SME WN 15k (d=300)	39.0	49.7	60.0	20.0	32.1	40.3
SME WN 30k (d=300)	40.2	67.8	63.8	32.7	37.9	43.9
SME WN 45k (d=300)	42.0	51.9	59.1	21.8	34.7	43.3
SME WN 60k (d=300)	39.6	57.7	54.9	26.2	39.7	42.4
SME WN 90k (d=300)	37.3	67.0	54.3	26.1	37.0	45.4
GloVe	40.8	76.9	80.4	69.3	80.6	71.6

Table A.2: Results of the intrinsic evaluation of the SME (edge reconstruction based) models for increasing size of the WordNet subgraph (15-90k) and two embedding dimensions: 50 and 300; and the text-based model, GloVe. Presented scores (rows) are Spearman's rank-order correlation coefficients of the obtained vector similarities against the gold standard defined by each of the six testsets (columns).

	Similarity			Relatedness		
	Simlex-999	**RG1965**	**WS353-Sim**	**WS353-Rel**	**MEN**	**MTurk-771**
MF SWOW R1 (d=300)	64.4	88.6	83.4	77.5	85.8	80.7
MF SWOW R1 (d=850)	65.6	88.8	83.2	77.6	85.6	81.2
MF SWOW R123 (d=300)	67.8	92.9	85.0	79.3	87.2	80.9
MF SWOW R123 (d=850)	70.0	93.2	85.3	80.6	87.5	82.3
RW SWOW (d=300)	69.3	90.2	84.5	77.7	84.3	81.1
ER SWOW (d=300)	54.1	83.5	77.1	70.7	78.5	74.8
GloVe	40.8	76.9	80.4	69.3	80.6	71.6

Table A.3: Results of the intrinsic evaluation of the models based on the SWOW graph: 1) four models using matrix factorisation (MF): two based on relation *R1* and two based on relation *R123*, using embedding dimensions of 300 and 850; 2) random walk based model (RW); 3) edge reconstruction based model (ER); 4) text-based model, GloVe. Presented scores (rows) are Spearman's rank-order correlation coefficients of the obtained vector similarities against the gold standard defined by each of the six testsets (columns).

	Similarity			Relatedness		
	Simlex-999	**RG1965**	**WS353-Sim**	**WS353-Rel**	**MEN**	**MTurk-771**
MF WN	49.9	57.0	50.8	30.9	45.0	52.8
RW WN	50.9 ± 0.2	77.5 ± 1.0	67.4 ± 0.3	28.4 ± 0.8	52.2 ± 0.7	52.9 ± 0.5
ER WN	39.6 ± 1.6	57.7 ± 4.8	54.9 ± 2.3	26.2 ± 4.1	39.7 ± 2.6	42.4 ± 1.3
MF SWOW	67.8	92.9	85.0	79.3	87.2	80.9
RW SWOW	69.3 ± 0.1	90.2 ± 0.5	84.5 ± 0.1	77.7 ± 0.2	84.3 ± 0.1	81.1 ± 0.2
ER SWOW	54.1 ± 6.2	83.5 ± 4.5	77.1 ± 4.8	70.7 ± 3.7	78.5 ± 3.9	74.8 ± 4.2
GloVe	40.8	76.9	80.4	69.3	80.6	71.6
Random	3.7 ± 8.7	5.7 ± 9.9	6.2 ± 1.0	3.5 ± 5.7	0.3 ± 1.7	2.6 ± 2.2

Table A.4: Results of the intrinsic evaluation of the 8 models for comparison. All WordNet models arebased on the same vocabulary subset (60k). The embedding dimension is 300. Presented scores (rows) are Spearman's rank-order correlation coefficients of the obtained vector similarities against the gold standard defined by each of the six testsets (columns). The deviation from averaging over three runs is indicated where relevant. The values in *Random* row stand for scores from the baseline of randomly initialised vectors.

Appendix B. Complete results of the extrinsic evaluation

	CoLA	SST-2	MRPC	RTE	WNLI
MF WN	14.97 ± 0.72	74.73 ± 0.46	80.17 ± 0.65	55.87 ± 0.91	45.53 ± 10.75
RW WN	11.57 ± 3.82	79.13 ± 1.27	79.85 ± 1.30	57.17 ± 0.57	39.00 ± 8.26
ER WN	12.47 ± 0.55	68.80 ± 0.82	77.98 ± 0.68	55.70 ± 0.56	49.30 ± 6.42
MF SW	8.30 ± 1.71	79.67 ± 0.57	78.78 ± 0.57	55.73 ± 1.44	40.83 ± 5.10
RW SW	15.90 ± 1.15	80.87 ± 0.42	80.03 ± 0.32	55.23 ± 0.75	38.00 ± 1.40
ER SW	8.80 ± 1.67	69.47 ± 1.04	76.90 ± 0.09	54.63 ± 1.80	46.50 ± 4.85
Glove	13.70 ± 1.95	82.57 ± 0.85	79.62 ± 1.42	58.27 ± 0.81	50.23 ± 5.66
Random	13.70 ± 2.50	63.33 ± 0.71	77.67 ± 0.47	57.80 ± 1.80	47.87 ± 8.57

Table B.1: Results of the extrinsic evaluation on selected GLUE tasks (columns) of the models using different types of pretrained embeddings as input (rows). Performance is measured in the following metrics: Matthews Correlation Coefficient for CoLA, the average of accuracy and F1-score for MRPC and accuracy for the remaining tasks. For clarity, the scores are adapted to the interval of [0-100].

Appendix C. Complete results of the evaluation using the diagnostic set

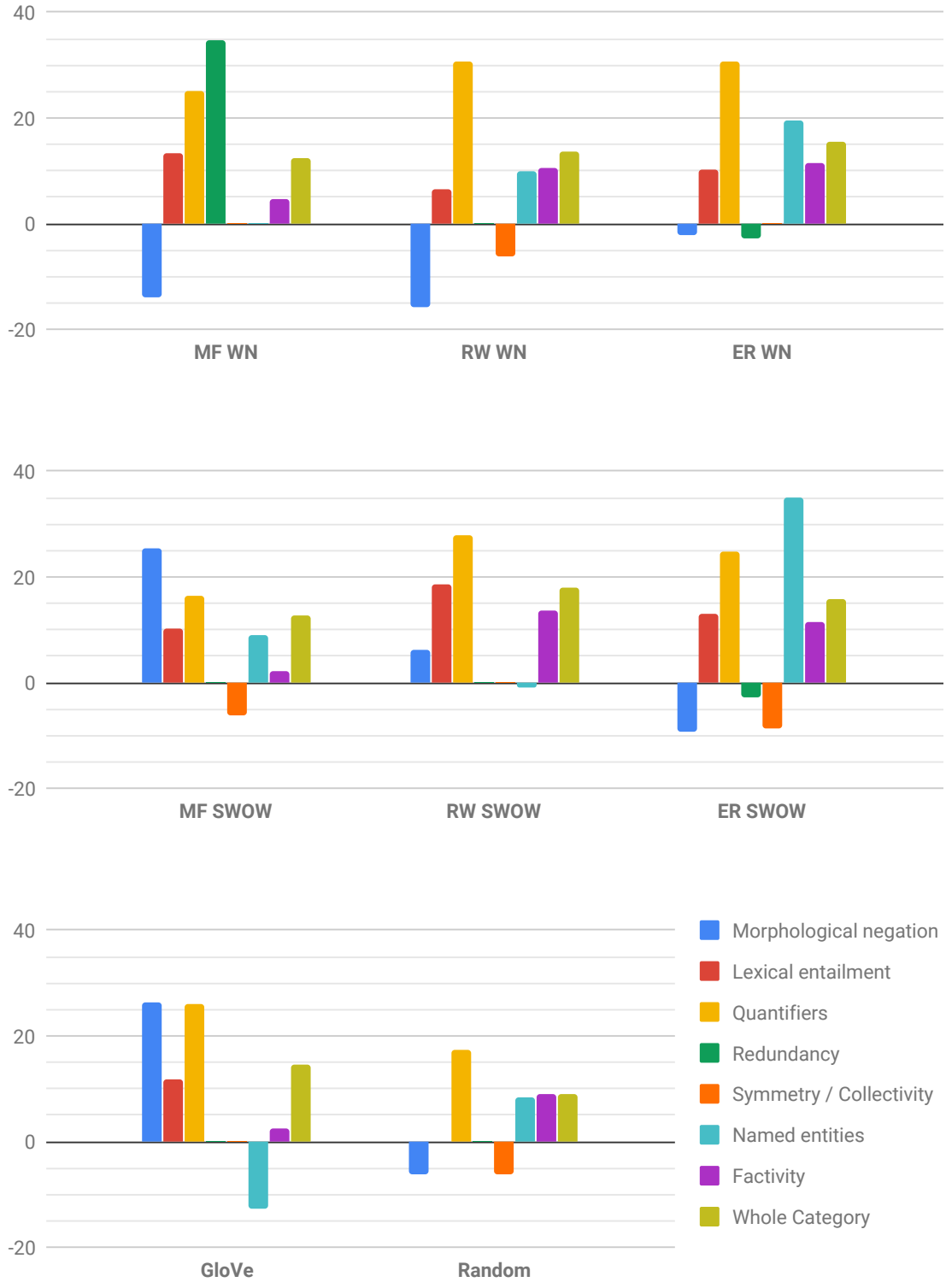

Figure C.1: Scores for the Lexical Semantics category in the Diagnostic dataset for all 8 models.

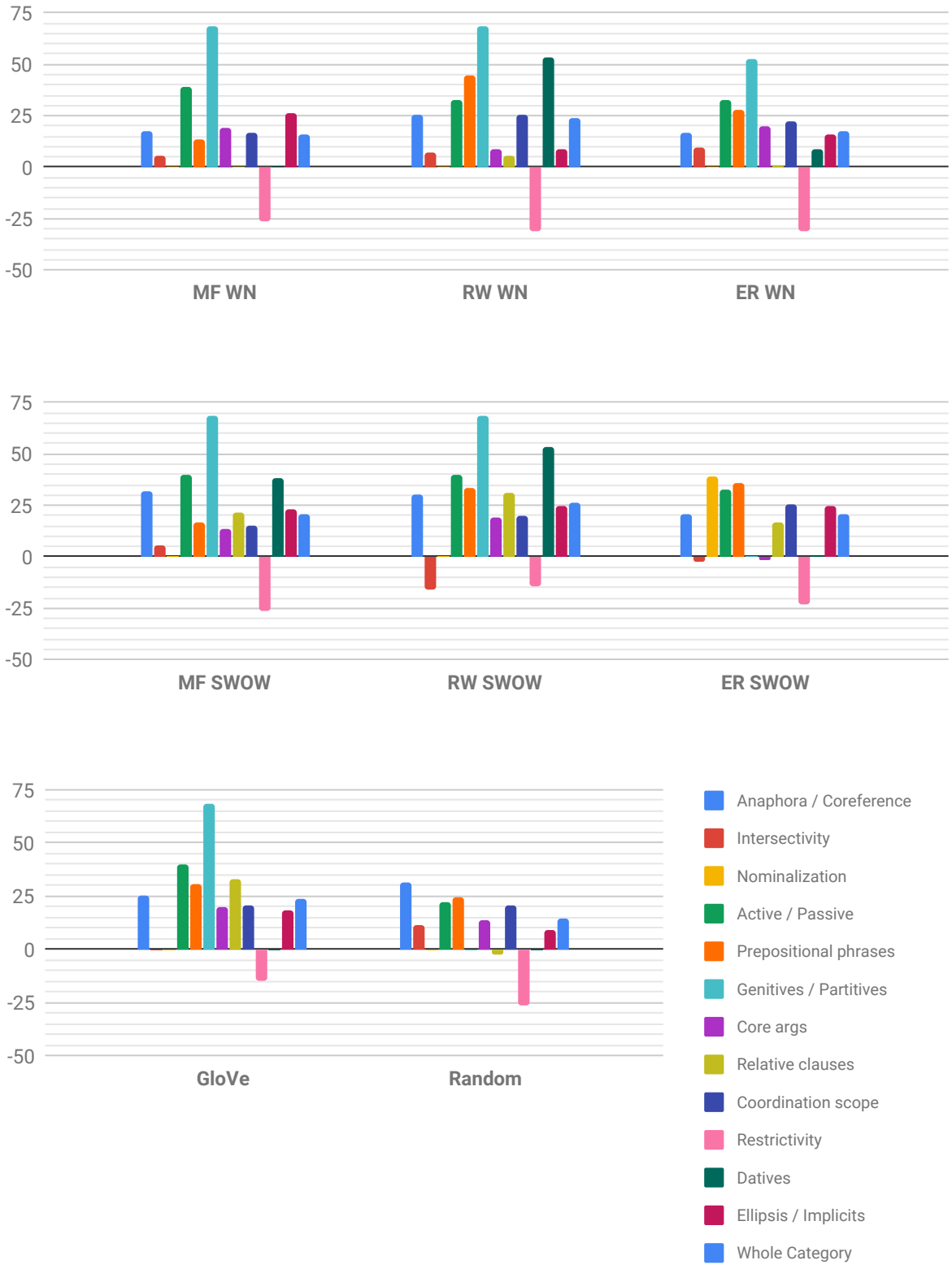

Figure C.2: Scores for the Predicate-Argument Structure category in the Diagnostic dataset for all 8 models.

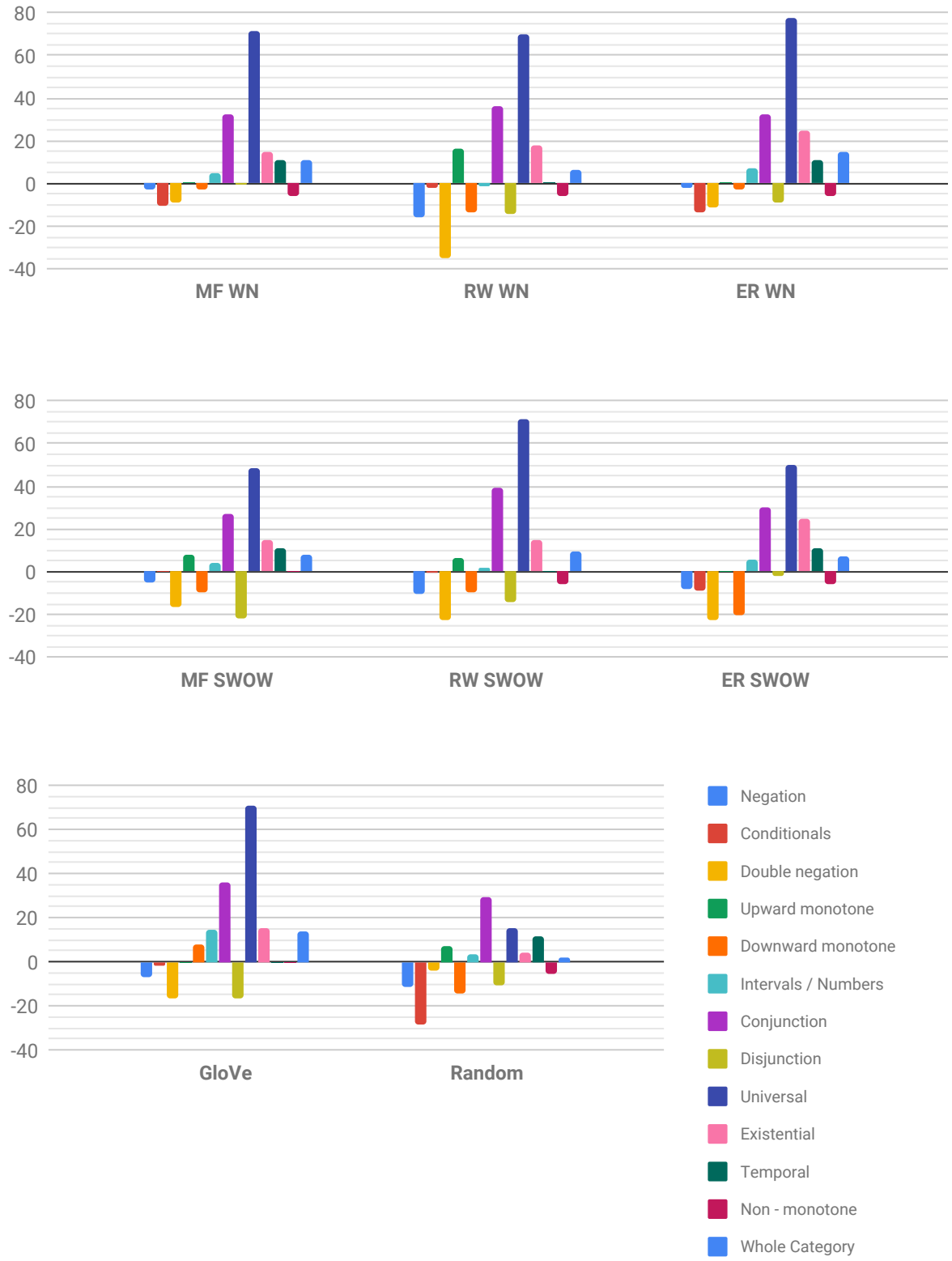

Figure C.3: Scores for the Logic category in the Diagnostic dataset for all 8 models.

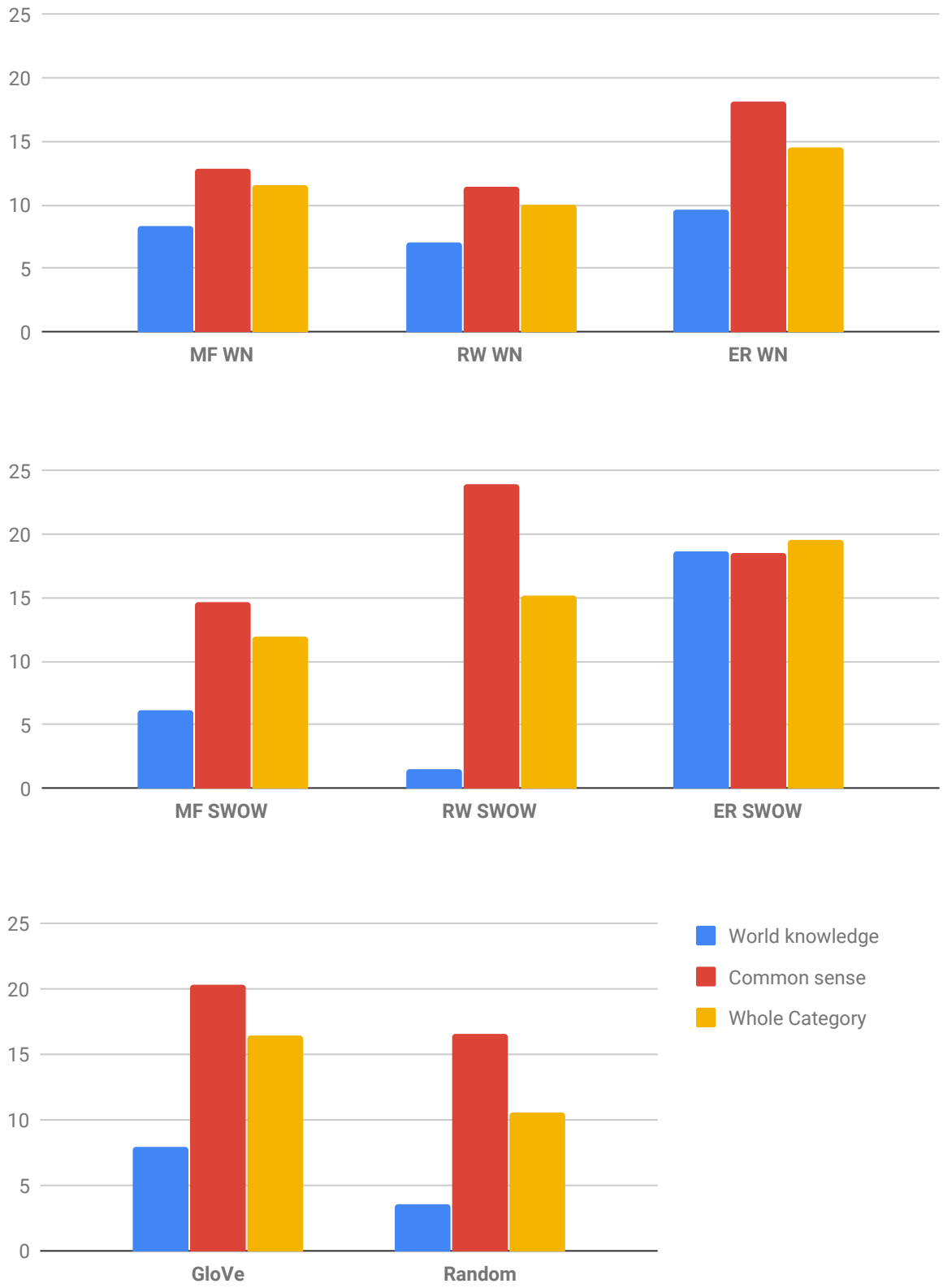

Figure C.4: Scores for the Knowledge category in the Diagnostic dataset for all 8 models.

Bibliography

[1] Samuel R Bowman, Ellie Pavlick, Edouard Grave, Benjamin Van Durme, Alex Wang, Jan Hula, Patrick Xia, Raghavendra Pappagari, R Thomas McCoy, Roma Patel, et al. Looking for elmo's friends: Sentence-level pretraining beyond language modeling. *arXiv preprint arXiv:1812.10860*, 2018.

[2] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[3] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019. URL: `https://openreview.net/forum?id=rJ4km2R5t7`.

[4] M Ross Quillan. Semantic memory. Technical report, Bolt Beranek and Newman Inc., Cambridge MA, 1966.

[5] George A Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[6] Marvin Minsky. A framework for representing knowledge. In *Psychology of Computer Vision*. McGraw-Hill, 1975.

[7] Daniel G. Bobrow and Donald Arthur Norman. Some principles of memory schemata. In *Representation and Understanding: Studies in Cognitive Science*, page 131–149. Elsevier, 1975.

[8] Simon De Deyne, Danielle J Navarro, Amy Perfors, Marc Brysbaert, and Gert Storms. The "small world of words" english word association norms for over 12,000 cue words. *Behavior research methods*, pages 1–20, 2018.

[9] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[10] Charles E Osgood, George J Suci, and Percy H Tannenbaum. The measurement of meaning. *Urbana: University of Illinois Press*, 1957.

[11] L Wittgenstein. In gem anscombe. *Philosophical investigations*, 1953.

[12] Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. Semantic similarity from natural language and ontology analysis. *Synthesis Lectures on Human Language Technologies*, 8(1):1–254, 2015.

[13] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25, 2000.

[14] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[17] Hongyun Cai, Vincent W Zheng, and Kevin Chang. A comprehensive survey of graph embedding: problems, techniques and applications. *IEEE Transactions on Knowledge and Data Engineering*, 2018.

[18] Mark Newman. *Networks: an introduction*. Oxford university press, 2010.

[19] Chakaveh Saedi, António Branco, João António Rodrigues, and João Silva. Wordnet embeddings. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 122–131. Association for Computational Linguistics, 2018. URL: `http://aclweb.org/anthology/W18-3016`.

[20] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[21] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.

[22] Josu Goikoetxea, Aitor Soroa, and Eneko Agirre. Random walks and neural network language models on knowledge bases. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics and Human Language Technologies Conference (NAACL-HLT25)*, pages 1434–1439. Association for Computational Linguistics, 2015.

[23] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259, 2014.

[24] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[25] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[26] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[27] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[28] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. *ACM Transactions on information systems*, 20(1):116–131, 2002.

[29] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 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, 2009.

[30] Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, 2015.

[31] Herbert Rubenstein and John B Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

[32] Elia Bruni, Nam-Khanh Tran, and Marco Baroni. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47, 2014.

[33] Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 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, 2012.

[34] Alex Wang, Ian F. Tenney, Yada Pruksachatkun, Katherin Yu, Jan Hula, Patrick Xia, Raghu Pappagari, Shuning Jin, R. Thomas McCoy, Roma Patel, Yinghui Huang, Jason Phang, Edouard Grave, Najoung Kim, Phu Mon Htut, Thibault F'evry, Berlin Chen, Nikita Nangia, Haokun Liu, , Anhad Mohananey, Shikha Bordia, Ellie Pavlick, and Samuel R. Bowman. jiant 0.9: A software toolkit for research on general-purpose text understanding models. `http://jiant.info/`, 2019.

[35] Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*, 2018.

[36] Brian W Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451, 1975.

[37] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[38] William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

[39] Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer, 2005.

[40] Roy Bar-Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. The second pascal recognising textual entailment challenge. In *Proceedings of the second PASCAL challenges workshop on recognising textual entailment*, volume 6, pages 6–4. Venice, 2006.

[41] Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9. Association for Computational Linguistics, 2007.

[42] Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The fifth pascal recognizing textual entailment challenge. In *TAC 2009 Workshop*, 2009.

[43] Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2012.

[44] Jan Gorodkin. Comparing two k-category assignments by a k-category correlation coefficient. *Computational biology and chemistry*, 28(5-6):367–374, 2004.

[45] Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics. URL: `https://www.aclweb.org/anthology/P18-1198`.