Grammatical Role Embeddings for Enhancements of Relation Density in the Princeton WordNet

1 Introduction

Word Embeddings play an important role in the new stream of natural language processing applications, providing latent features for lexical items. In many cases it is expected that the necessary features are encoded within the embedding space. Unfortunately, we may check this only indirectly.

In the paper we report embeddings on the subatom level[1] that make explicit some of the features related to the semantic selectional restrictions on grammatical roles of words in text. Thus our goal is not to learn an embedding for a verb, but rather embeddings for the participants in the event (or state) denoted by that verb.

We exploit these embeddings for the addition of new syntagmatic relations to WordNet — (Fellbaum, 1998) — with the aim to improve applications such as knowledge-based word sense disambiguation (KWSD). Evaluation in the paper is performed by automatically extending WordNet with ranked relations within the context of KWSD. However, the ranking over the candidate relations might also be exploited for manual inspection and validation of the new relations.

The structure of the paper is as follows: Section 2 briefly discusses related work. Section 3 presents a motivating example of sentence semantics based on Minimal Recursion Semantics. The section 4 describes the mechanism for creation of grammatical role embeddings. In section 5 we present our goal to extend WordNet with syntagmatic relations. In section 6 the experiment setup is presented and the results are discussed. The last section concludes the paper.

[1]By subatom level we mean the elements of a predicate.

2 Related Work

The success of KWSD approaches apparently depends on the quality of the knowledge graph – whether the knowledge represented in terms of nodes and relations (arcs) between them is sufficient for the algorithm to pick the correct senses of ambiguous words. Several extensions of the knowledge graph constructed on the basis of WordNet have been proposed and implemented. With respect to the extension of WordNet with syntagmatic information also there are many works such as (Bentivogli and Pianta, 2004) and (Lothar Lemnitzer and Gupta, 2008).

Here we present in details only one approach similar to the one presented here is described in Agirre and Martinez (2002), which explores the extraction of syntactically supported semantic relations from manually annotated corpora. In this line of research SemCor (Miller et al., 1993), as a semantically annotated corpus, was processed with the MiniPar dependency parser and the subject-verb and object-verb relations were consequently extracted. The new relations were represented on several levels: as word-to-class and class-to-class relations. The extracted selectional relations were then added to WordNet and used in the WSD task. The main differences with the presently described approach are: the set of relations used here is bigger (it includes also indirect-object-to-verb relations). Also, the new relations in the present paper are not added as selectional relations, but as semantic relations between the corresponding synsets. This means that the specific syntactic role of the participant is not taken into account, but only the connectedness between the participant and the event is registered in the knowledge graph. Also, in our work we use embeddings as filters, instead of the selectional restrictions approach in Agirre and Martinez (2002).

There is also a huge number of works related to extending of World knowledge related Knowledge Graphs with new relations such as (Minervini et al., 2015). We will not present any of them now, but in the final version an overview will be presented.

3 Minimal Recursion Semantics

In many semantic theories lexical semantics is represented by using not just predicates from first order logic, but a more complicated schema which would allow access to more detailed representation of the semantic interpretations. As an illustration of such a kind of semantics we assume Minimal Recursion Semantics (MRS). Here we present just one motivating example of an MRS structure for the sentence *"Every dog chases some white cat."*

<h0, {h1:every(x,h2,h3), h2:dog(x),
h4:chase(e, x, y), h5:some(y,h6,h7),
h6:white(y), h6:cat(y)}, {}>

The top handle is h0. The quantifiers are represented as the relations $every(x, y, z)$ and $some(x, y, z)$, where x is the bound variable, y and z are handles determining the restriction and the body of the quantifier. The conjunction of two or more relations is represented by sharing the same handle (h6 above). The outscope relation is defined as a transitive closure of the immediate outscope relation between two elementary predications — EP immediately outscopes EP' iff one of the scopal arguments of EP is the label of EP'. In the example the set of handle constraints is empty, which means that the representation is underspecified with respect to the scope of both quantifiers.

In order to use word embeddings over MRS structures we need to determine the interactions of the latent features for each of the predicate arguments. For example, the features from the embeddings for 'every', 'dog', and 'chase' have to agree on the common argument denoted by the variable 'x'.

In order to control this interaction in a better way we would like for each multiargument predicate to learn an embedding per argument. Thus for the above MRS structure we will need to have embeddings for 'x', 'y', 'e', 'h0', ... 'h7'. When we have them, we would like also to create an embedding related to the first argument of 'every'. The argument of 'dog' and the second argument of 'chase' have to "agree".

Our long-term goal is to train such subatom embeddings. In order to achieve this goal we assume that an extension of lexical resources, such as WordNet, with additional knowledge (in sense of relations) is necessary. Here we present an approach for learning such embeddings for grammatical roles. Then we use the grammatical role embeddings for extending of WordNet with syntagmatic relations.

4 Grammatical Role Embeddings from Parsed Corpora

In our first experiment we learned subatom semantic embeddings on the basis of dependency-parsed corpora. We determined the arguments as wordforms in the text. As an example, for the above mentioned case we used the position of 'dog'. In order to generalize over the different word forms in the different examples in the corpus we substituted the wordforms for the corresponding argument with a pseudoword form. For example, in the above sentence we generated the following variations with pseudoword forms for the different arguments of the different predicates:

Every *SUBJ-CHASE* chases some white cat.

Every dog chases some white *OBJ-CHASE*.

and many more. Having learned embeddings for these pseudowords, we assume that they represent the selectional features for the corresponding grammatical roles of the verbs.

5 WordNet Extensions with New Relations

As mentioned above, our goal is to extend WordNet with syntagmatic relations using semantically annotated corpora such as SemCor. If we have a subject-verb syntactic relation in the corpus, and the related verb and noun are manually annotated with synset ids from WordNet, we reliably assume that there is a semantic relation between the noun and the verb synsets in WordNet. At a more general level we call this relation `has-participant` relation that is directed from the verb to the noun synset. In order to draw a distinction between the different participants in an event (state), we use subrelations named after their grammatical roles: `has-subj`, `has-dobj`, and `has-iobj`.[2]

Adding a `has-participant` relation between two synsets in WordNet imposes two questions: (1) Does this relation hold for more specific synsets? (2) Does this relation generalize to more general synsets? In our previous research on extending WordNet with new relations from semantically and syntactically annotated corpora — (see XXX) — we have shown that using inference ... . Especially with respect to the `has-participant` relation, we assume that the relation holds when the noun synset is substituted with a hyponymic synset and that it also holds when the verb synset is substituted with a hypernymic synset. In this paper we consider only inference over noun synsets. With respect to generalization, we have to select noun synsets outside of the sub-hierarchy of the initial noun synset. Here we apply two approaches to generalization: (1) We select the direct hypernym assuming that the relation holds for it. Then we perform inheritance inference, assuming that the relation holds also for it hyponymic synsets; (2) We check all noun synsets as potential arguments of the relation. The first approach provide a better guess, but it could miss some generalizations. The second one is better for verbs that do not appear in the annotated corpora.

One of the problems we had is related to the the extension of WordNet with relations resulting from inference over the WordNet structure, such as the inheritance of relations. For example, we assume that if we had a relation for an agent of a verb like: "A doctor kisses a girl", then any kind of doctor could kiss a girl. For many predicates, however, this inference rule would not be correct. For example, "A doctor operates a patient" does not entail that all doctors can operate. In order to check which ones are appropriate, we used subatom semantic embeddings.

[2]In future work we plan to switch to semantic role names.

6 Experiments and Results

In this section we describe the experimental set up and the results.

Corpora preparation.

The corpora that the algorithms for word embeddings are trained on can contain either natural language text (e.g. Wikipedia or newswire articles) or artificially generated pseudo texts. Such pseudo texts can be the output from the Random Walk algorithm, when it is set to the mode of selecting sequences of nodes from a knowledge graph (KG) — see (Goikoetxea et al., 2015) for generation of pseudo corpora from a WordNet knowledge graph and (Ristoski and Paulheim, 2016) for generation of pseudo corpora from RDF knowledge graphs such as DBPedia, GeoNames, FreeBase. Here we report results only for knowledge graphs based on WordNet and its extensions.

The corpus for training of the embeddings reported in the paper consists of two parts: (1) pseudo corpus generated over WordNet (PCWN); and (2) real text corpora (RTC). PCWN is used to ensure that the embeddings represent features extracted from the knowledge within the WordNet. RTC is used to represent relevant contexts for learning embeddings of pseudo words for subjects, direct objects and indirect objects.

As RTC we have used WaCkypedia_EN corpus — (Baroni et al., 2009). The WaCkypedia_EN corpus was reparsed with a more recent version of the Stanford CoreNLP dependency parser. The dependency of type "collapsed-cc" was selected, which collapses several dependency relations in order to obtain direct dependencies between content words, and in addition propagates dependencies involving conjuncts. For instance, a parse of the sentence "the dog runs and barks" would result in the relations nsubj(dog, runs) and nsubj(dog, barks). This type of dependency allows for a token to have multiple head words.

The head word of each noun phrase subject, as well as direct and indirect object, is then replaced by its predicate role and its governing verb's lemma (SUBJ_run, SUBJ_bark — both for the noun 'dog'). When a token has more than one head word suitable for substitution, copies of the sentence are created for each alternative replacement.

For the relation `has-subj` we use the dependency relations 'nsubj' and 'nsubjpass'; for the relation `has-dobj` we use the dependency relation 'dobj'; and for the relation `has-iobj` we use the dependency relation 'iobj'. In order to minimize some errors we enforced a condition that the dependency word should be a noun.

The union of both corpora is used in the experiments. In RTC all the words were substituted with their lemmas. Punctuation marks and numbers were deleted. We used the Word2Vec tool[3] in order to train the embeddings. From the various models we select the one with the best score on the similarity task. This model was trained with the following settings: context window of 5 words; 7 iterations; negative examples set to 5; and frequency cut sampling set to 7.

Selection of candidate relations.

For the selection of candidate relations to be added to WordNet, we use the sense annotations in SemCor. The corpus comprises texts from the Brown corpus[4], which is a balanced corpus resource. Due to that, SemCor contains very diverse text genres. We use SemCor in two ways: first, for testing the output of the KWSD system for data in English — see below; and second, as a source for extracting of new semantic relations. To achieve this, we use the parsed SemCor. In our earlier work we had used a version of SemCor processed with a dependency parser included in the IXA pipeline[5]. We divided the corpus in a proportion one-to-three: first part comprised 49 documents (from br-a01 to br-f44) and it was used as a test set in the experiments reported below in the paper. The rest of the documents formed the training set from which the new relations were extracted.

The extracted relations are represented in the following format:

u:noun-synset-id v:verb-synset-id

Here is an example:

u:00031264-n v:02005948-v

for 'arrive' (02005948-v) and 'group' for (00031264-n). All the relations extracted in this way from SemCor are assumed to be correct. In addition to them we also used inferred relations as it was described above.

Ranking

Here we provide a description of the algorithm for ranking candidates for new relations. Let us assume that we are working with one type of relation, e.g. the SUBJ relation. From the available syntactic relations (extracted from SemCor), we select subject-verb relations. We use those in order to construct new relations via inheritance. That is, we fix the verb synset node and then expand the new relations along the noun subhierarchy of WordNet (which in the example case would mean expanding new relations between the 'arrive' synset and hyponyms of the 'group' synset); or in the second case, whereby we examine all noun synsets possible, we simply start the inheritance procedure from a top node in the hierarchy. From the new candidate relations, we select such that have noun nodes that are semantically similar to the relevant SUBJ-verb embeddings (i.e. all noun synsets whose embeddings satisfy a similarity condition with respect to the embedding of the participant for that verb).

Since the synset embeddings are not directly available (the embeddings reported in section 4 are trained only on lemmatized text, not on synset IDs), we need to calculate those. Thus, for each synset, we obtain its vector by averaging the vectors for all lemmas it can be expressed with (this information is retrieved from WordNet). For grammatical roles, we average the corresponding grammatical role vectors per each lemma in the particular verb synset; in this way, if a particular synset comprises N lemmas, we will average the vectors for *SUBJ_lemma1, SUBJ_lemma2, ..., SUBJ_lemmaN.*

After this step is completed, we have all the information necessary to compare synset embeddings with grammatical role embeddings that match verb synsets. The comparison is carried out by calculating the cosine similarity measure. By setting a similarity threshold, the filter can be controlled, so that more or fewer new relations are added to the extended graph. The same procedure is repeated for DOBJ and IOBJ relations.

Using this approach for each candidate rela-

[3]https://code.google.com/archive/p/word2vec/

[4]`http://clu.uni.no/icame/manuals/BROWN/INDEX.HTM`

[5]`http://ixa.si.ehu.es/Ixa`

tion we calculate the cosine similarity measure between the noun synset embedding and the embedding for the corresponding grammatical role. We are using the result as a rank of the candidate relations.

Experiments with Knowledge-based Word Sense Disambiguation.

In order to check the usefulness of the added relations, we performed experiments with the UKB system[6] for knowledge-based word sense disambiguation. The UKB tool requires two resource files to annotate the input text — a dictionary file with all lemmas that can be possibly linked to a sense identifier. In our case WordNet-derived relations were used for our knowledge base; consequently, the sense identifiers are WordNet IDs. For instance, a line from the WordNet extracted dictionary looks like this:

```
predicate 06316813-n:0 06316626-n:0
          01017222-v:0 01017001-v:0
          00931232-v:0
```

First comes the lemma associated with the relevant word senses, after the lemma the sense identifiers are listed. Each `ID` consists of eight digits followed by a hyphen and a label referring to the POS category of the word. Finally, a number following a colon indicates the frequency of the word sense, calculated on the basis of a tagged corpus. When a lemma from the dictionary has occurred in the analysis of the input text, the tool assigns all the associated word senses to the word form in the context and attempts to disambiguate its meaning among them.

The second resource file required for running the tool is the set of relations that is used to construct the knowledge graph over which UKB is run. The distribution of UKB comes with a file containing the standard lexical relations defined in WordNet, such as hypernymy, meronymy, etc., as well as with a file containing relations derived on the basis of common words found in the synset glosses, which have been manually disambiguated. The format of the relations in the KG is as follows:

```
u:SynSetId01 v:SynSetId02 s:Source d:w
```

where `SynSetId01` is the identifier of the first synset in the relation, `SynSetId02` is the identifier of the second synset, `Source` is the source of the relation, and `w` is the weight of the relation in the graph. In the experiments reported in the paper, the weight of all relations is set to 0.

In our experiments we relied on the following knowledge graphs: **wn30** — a knowledge graph formed from the relations in the WordNet, **wn30-syn** — a knowledge graph formed on the basis of **wn30** extended by the syntactic-based relations from SemCor, **wn30-synI** — a knowledge graph formed on the basis of **wn30-syn** extended by inferred relations, **wn30-synI40** — similar to the previous but all relations ranked under 0.4 are excluded, **wn30-synI45** — relations ranked under 0.45 are excluded, **wn30-synI50** — relations ranked under 0.5 are excluded. The knowledge graph **wn30-synIU** includes the relations from **wn30-syn** to which relations inferred by moving one hyperonym up and the inheritance down (see the descrption above), **wn30-synIU40** — similar to the previous, but the inferred relations ranked under 0.4 are excluded, **wn30-synIU45** — similar to the previous, but the inferred relations ranked under 0.45 are excluded, **wn30-synIU50** — similar to the previous, but the inferred relations ranked under 0.5 are excluded. The knowledge graph **wn30-cwn-45-semc6-50** was created in order to check the idea for constructing relations using all noun synsets as potential arguments of the relation. For this knowledge we include all the verbs from CoreWordNet and the most frequent verbs from the training part of SemCor. For each verb we checked more than 70 000 possible relations. Here we report the best result in which we used all the relations for verbs from CoreWordNet ranked above 0.45 and all the relations for verbs from SemCor ranked above 0.5.

The evaluation of the Word Sense Disambiguation is done over two test data sets: the test part of SemCor (see above) and the English part of the test data set for the Multilingual Word Sense Disambiguation[7] — named here M13 SemeVal. The results are presented in Table 1. For SemCor there are improvements after filtering for both cases: (1) exten-

[6] `http://ixa2.si.ehu.es/ukb/`

[7] `https://www.cs.york.ac.uk/semeval-2013/task12/`

Knowledge Graph	SemCor	M13 SemeVal
wn30	51.56	**48.41**
wn30-syn	55.74	*45.74*
wn30-synI	54.15	45.92
wn30-synI40	**55.99**	45.13
wn30-synI45	55.60	46.04
wn30-synI50	55.71	**46.16**
wn30-synIU	52.53	**46.53**
wn30-synIU40	**56.22**	46.28
wn30-synIU45	55.50	**46.53**
wn30-synIU50	55.76	46.47
wn30-cwn-45-semc6-50	51.35	**48.84**

Table 1: Results for Word Sense Disambiguation for the different Knowledge Graphs.

sion for the relations from syntax; and (2) generalization to first hyperonym and inheritance down in the hierarchy. For M13 SemeVal the addition of syntactic relations causes a drop in the performance. Our explanation of this fact is that the two datasets are in very different domains. Still, after the drop, the application of inference and filtering improves the result. In the knowledge graph **wn30-cwn-45-semc6-50** we note a small drop in the case of SemCor, and improvement in the case of M13 SemeVal. Our explanation here is that the huge number of candidate relations makes the selectional filter hard to tune.

On the basis of these results we conclude that the proposed mechanism of adding syntagmatic relations to WordNet is feasible. We are in process of manual checking the ranked relations, and provided that the paper is accepted the final version will report on this.

7 Conclusion

This paper presents an approach for learning features by subtomic semantic representation. Our longterm plan is to design a learning approach for each semantic argument in an MRS representation. The results here are the first steps in this direction. We plan to proceed as follows: (1) process the training corpus on more levels, including part-of-speech annotation, semantic annotation with word senses from WordNet; (2) include more arguments in the learning process like arguments of relational nouns and adjectives, which will impose mutual constraints on the learned features; (3) perform tuning to the linguistic knowledge already represented in WordNet, FrameNet and other lexical resources and manually annotated corpora, by techniques similar to retrofitting. The learned features are used to enrich WordNet with new syntagmatic relations. For extrinsic evaluation we used the KWSD task.

References

Eneko Agirre and David Martinez. 2002. Integrating selectional preferences in wordnet. In *Proceedings of First International WordNet Conference.*

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Luisa Bentivogli and Emanuele Pianta. 2004. Extending wordnet with syntagmatic information. In *In Second Global WordNet Conference*, pages 47–53.

Christiane Fellbaum, editor. 1998. *WordNet An Electronic Lexical Database.* The MIT Press, Cambridge, MA ; London, May.

Josu Goikoetxea, Aitor Soroa, and Eneko Agirre. 2015. Random walks and neural network language models on knowledge bases. In *HLT-NAACL*, pages 1434–1439. The Association for Computational Linguistics.

Holger Wunsch Lothar Lemnitzer and Piklu Gupta. 2008. Enriching germanet with verb-noun relations - a case study of lexical acquisition. In Bente Maegaard Joseph Mariani Jan Odijk Stelios Piperidis Daniel Tapias

Nicoletta Calzolari (Conference Chair), Khalid Choukri, editor, *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

George A. Miller, Claudia Leacock, Randee Tengi, and Ross T. Bunker. 1993. A Semantic Concordance. In *Proceedings of the workshop on Human Language Technology*, HLT '93, pages 303–308, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pasquale Minervini, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito. 2015. Efficient learning of entity and predicate embeddings for link prediction in knowledge graphs. In *Proceedings of the 11th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2015) co-located with the 14th International Semantic Web Conference (ISWC 2015), Bethlehem, USA, October 12, 2015.*, pages 26–37.

Petar Ristoski and Heiko Paulheim, 2016. *RDF2Vec: RDF Graph Embeddings for Data Mining*, pages 498–514. Springer International Publishing, Cham.