Replacing OOV Words For Dependency Parsing With Distributional Semantics

Prasanth Kolachina [△] and **Martin Riedl** [◇] and **Chris Biemann** [◇]
[△] Department of Computer Science and Engineering, University of Gothenburg, Sweden
[◇] Language Technology Group, Universität Hamburg, Germany
prasanth.kolachina@gu.se
{riedl,biemann}@informatik.uni-hamburg.de

Abstract

Lexical information is an important feature in syntactic processing like part-of-speech (POS) tagging and dependency parsing. However, there is no such information available for out-of-vocabulary (OOV) words, which causes many classification errors. We propose to replace OOV words with in-vocabulary words that are semantically similar according to distributional similar words computed from a large background corpus, as well as morphologically similar according to common suffixes. We show performance differences both for count-based and dense neural vector-based semantic models. Further, we discuss the interplay of POS and lexical information for dependency parsing and provide a detailed analysis and a discussion of results: while we observe significant improvements for count-based methods, neural vectors do not increase the overall accuracy.

1 Introduction

Due to the high expense of creating treebanks, there is a notorious scarcity of training data for dependency parsing. The quality of dependency parsing crucially hinges on the quality of part-of-speech (POS) tagging as a preprocessing step; many dependency parsers also utilize lexicalized information, which is only available for the training vocabulary. Thus errors in dependency parsers often relate to OOV (out of vocabulary, i.e. not seen in the training data) words.

While there has been a considerable amount of work to address the OOV problem with continuous word representations (see Section 2), this requires a more complex model and hence, increases training and execution complexity.

In this paper, we present a very simple yet effective way of alleviating the OOV problem to some extent: we use two flavors of distributional similarity, computed on a large background corpus, to replace OOV words in the input with semantically or morphologically similar words that have been seen in the training, and project parse labels back to the original sequence. If we succeed in replacing OOV words with in-vocabulary words of the same syntactic behavior, we expect the tagging and parsing process to be less prone to errors caused by the absence of lexical information.

We show consistent significant improvements both for POS tagging accuracy as well as for Labeled Attachment Scores (LAS) for graph-based semantic similarities. The successful strategies mostly improve POS accuracy on open class words, which results in better dependency parses. Beyond improving POS tagging, the strategy also contributes to parsing accuracy. Through extensive experiments – we show results for seven different languages – we are able to recommend one particular strategy in the conclusion and show the impact of using different similarity sources.

Since our method manipulates the input data rather than the model, it can be used with any existing dependency parser without re-training, which makes it very applicable in existing environments.

2 Related Work

While part-of-speech (POS) tags play a major role in detecting syntactic structure, it is well known (Kaplan and Bresnan (1982) inter al.) that lexical information helps for parsing in general and for

Proceedings of the 21st Nordic Conference of Computational Linguistics, pages 11–19,
Gothenburg, Sweden, 23-24 May 2017.

dependency parsing in particular, see e.g. Wang et al. (2005).

In order to transfer lexical knowledge from the training data to unseen words in the test data, Koo et al. (2008) improve dependency parsing with features based on Brown Clusters (Brown et al., 1992), which are known to be drawing syntactic-semantic distinctions. Bansal et al. (2014) show slight improvements over Koo et al. (2008)'s method by tailoring word embeddings for dependency parsing by inducing them on syntactic contexts, which presupposes the existence of a dependency parser. In more principled fashion, Socher et al. (2013) directly operate on vector representations. Chen et al. (2014) address the lexical gap by generalizing over OOV and other words in a feature role via feature embeddings. Another approach for replacing OOV words by known ones using word embeddings is introduced by Andreas and Klein (2014).

All these approaches, however, require retraining the parser with these additional features and make the model more complex. We present a much simpler setup of replacing OOV words with similar words from the training set, which allows retrofitting any parser with our method.

This work is related to Biemann and Riedl (2013), where OOV performance of fine-grained POS tagging has been improved in a similar fashion. Another similar work to ours is proposed by Huang et al. (2014), who replace OOV named entities with named entities from the same (fine-grained) class for improving Chinese dependency parsing, which largely depends on the quality of the employed NER tagger and is restricted to named entities only. In contrast, we operate on all OOV words, and try to improve prediction on coarse universal POS classes and universal dependencies.

On a related note, examples for a successful application of OOV replacements is demonstrated for Machine Translation (Gangadharaiah et al., 2010; Zhang et al., 2012).

3 Methodology

For replacing OOV words we propose three strategies: replace OOV words by most similar ones using distributional semantic methods, replace OOV words with words with the most common suffix and replacing OOV words before or after POS tagging to observe the effect on dependency parsing. The influence of all components is evaluated separately for POS tagging and dependency parsing in Section 5.

3.1 Semantic Similarities

In order to replace an OOV word by a similar in-vocabulary word, we use models that are based on the distributional hypothesis (Harris, 1951). For showing the impact of different models we use a graph-based approach that uses the left- and right-neighbored word as context, represented by the method proposed by Biemann and Riedl (2013), and is called distributional thesaurus (*DT*). Furthermore, we apply two dense numeric vector-space approaches, using the skip-gram model (*SKG*) and CBOW model of the `word2vec` implementation of Mikolov et al. (2013).

3.2 Suffix Source

In addition, we explore replacing OOVs with words from the similarity source that are contained in the training set and share the longest suffix. This might be beneficial as suffixes reflect morphological markers and carry word class information in many languages. The assumption here is that for syntactic dependencies, it is more crucial that the replacement comes from the same word class than its semantic similarity. This also serves as a comparison to gauge the benefits of the similarity source alone. Below, these experiments are marked with *suffix*, whereas the highest-ranked replacement from the similarity sources are marked as *sim*. As a *suffix-only* baseline, we replace OOVs with its most suffix-similar word from the training data, irrespective of its distributional similarity. This serves as a sanity check whether semantic similarities are helpful at all.

3.3 Replacement Strategies regarding POS

We explore two different settings for dependency parsing that differ in the use of POS tags:

(1) *oTAG*: POS-tag original sequence, then replace OOV words, retaining original tags for parsing;

(2) *reTAG*: replace OOV word, then POS-tag the new sequence and use the new tags for parsing.

The *oTAG* experiments primarily quantify the sensitivity of the parsing model to word forms, whereas *reTag* assess the potential improvements in the POS tagging.

3.4 Replacement Example

As an example, consider the automatically POS-tagged input sentence "We/P went/V to/P the/D aquatic/N park/N" where "aquatic" is an OOV word. Strategy *oTAG sim* replaces "aquatic" with "marine" since it is the most similar in-vocabulary word of "aquatic". Strategy *oTAG suffix* replaces it with "exotic" because of the suffix "tic" and its similarity with "aquatic". The *suffix-only* baseline would replace with "automatic" since it shares the longest suffix of all in-vocabulary words. The *re-TAG* strategy would then re-tag the sentence, so the parser will e.g. operate on "We/P went/V to/P the/D marine/ADJ park/N". Table 1 shows an example for different similarity-based strategies for English and German[1]. We observe that the *sim* strategy returns semantically similar words that do not necessarily have the same syntactic function as the OOV target.

	sim	**sim&suffix**
English OOV: upgraded		
Suffix-only	paraded	
CBOW	upgrade	downloaded
SKG	upgrade	expanded
DT	expanded	updated
German OOV: Nachtzeit		
Suffix-only	Pachtzeit	
CBOW	tagsüber	Ruhezeit
SKG	tagsüber	Echtzeit
DT	Jahreswende	Zeit

Table 1: Here we show replacements for different methods using different strategies.

4 Experimental Settings

Here we describe the methods, background corpora used for computing similarities and all further tools used for the experiments. With our experiments, we target to address the following research questions:

- Can syntactic processing benefit from OOV replacement, and if so, under what strategies and conditions?
- Is there a qualitative difference between similarity sources with respect to tagger/parser performance?
- Are there differences in the sensitivity of parsing inference methods to OOV replacement?

4.1 Similarity Computations

We are using two different approaches to determine semantic similarity: a symbolic, graph-based framework for distributional similarity and a neural language model that encodes words in a dense vector space.

Graph-based Semantic Similarity

The computation of a corpus-based distributional thesaurus (marked as *DT* below) is performed following the approach by Biemann and Riedl (2013) as implemented in the JoBimText[2] software. For computing similarities between words from large unlabeled corpora, we extract as word-context the left and right neighboring words, not using language-specific syntactic preprocessing. Words are more similar if they share more of their most salient 1000 context features, where salient context features are ranked by Lexicographer's Mutual Information (LMI), (Evert, 2005). Word similarity in the DT is defined as the count of overlapping salient context features. In addition we prune similar words[3] below a similarity threshold of 5.

In order to use such a DT to replace an OOV word, we look up the most similar terms for the OOV word and choose the highest-ranked word from the training data vocabulary, respectively the most similar word with the longest common suffix.

Neural Semantic Similarity

As an alternative similarity we run `word2vec` with default parameters (marked as *w2v* below) (Mikolov et al., 2013) on our background corpora, obtaining 200-dimensional dense vector embeddings for all words with a corpus frequency larger than 5. We conduct this for both flavors of *w2v*: skipgram, marked as *SKG* below (based on positional windows) and *CBOW* (based on bag of word sentential contexts).

Following the standard approach, we use the cosine between word vectors as a similarity measure: for each OOV, we compare vectors from all words in the training set and pick the word that correspond to the most similar vector as a replacement,

[1] Translations: Nachtzeit = night time; tagsüber = during the day; Pachtzeit = length of lease; Ruhezeit = downtime; Echtzeit = real time; Jahreswende = turn of the year

[2] `http://www.jobimtext.org`

[3] we have tried a few thresholds in preliminary experiments and did not find results to be very sensitive in the range of 2 – 20

respectively the most similar word of those with the longest common suffix.

4.2 Corpora for Similarity Computation

As we perform the experiments on various languages, we will compute similarities for each language separately. The English similarities are computed based on 105M sentences from the Leipzig corpora collection (LCC) (Richter et al., 2006) and the Gigaword corpus (Parker et al., 2011). The German (70M) and the Hindi (2M) corpora are extracted from the LCC as well. We compute similarities on 19.7M sentences of Arabic, 259.7M sentences of French and 128.1M sentences of Spanish extracted from web corpora[4] provided by Schäfer and Bildhauer (2013). For the computation of the Swedish similarities we use a 60M-sentence news corpus from Spraakbanken.[5] In summary, all background corpora are in the order of about 1 Gigaword, except the Hindi corpus, which is considerably smaller.

4.3 Dependency Parser and POS Tagger

For the dependency parsing we use the implementation of the graph-based dependency parser provided in Mate-tools (Bohnet, 2010, version 3.6) and the transition-based Malt parser (Nivre, 2009, version 1.8.1). Graph-based parsers use global inference to construct the maximum spanning dependency tree for the input sequences. Contrary, the greedy algorithm in the transition-based parser uses local inference to predict the dependency tree. The parsing models for both parsers, Mate-tools and Malt parser, are optimized using cross-validation on the training section of the treebank[6]. We train the dependency parsers using POS tags (from the Mate-tools tagger) predicted using a 5-fold cross-validation. The evaluation of the parser accuracies is carried out using MaltEval. We report labeled attachment score (LAS) for both overall and on OOV token positions.

4.4 Treebanks

For training and testing we apply the treebanks (train/dev/test size in tokens in parentheses) from the Universal Dependencies project (Nivre et al., 2016, version 1.2 released November 15th, 2015) for Arabic, English, French, German, Hindi, Spanish and Swedish. Tagset definitions are available online.[7]

5 Results

In this section, we report experimental results and compare them to the baseline without OOV replacement. All statistical significance tests are done using McNemar's test. Significant improvements ($p < 0.05$) over the baseline without OOV replacement are marked with an asterisk (*), significant performance drops with a hashmark (#) and the best result per experiment is marked in bold.

5.1 Results for POS Tagging

In Table 2 we show overall and OOV-only POS tagging accuracies on the respective test set for seven languages using similarities extracted from the DT.

LANG	OOV	baseline		suffix only		DT sim		DT suffix	
	%	all	OOV	all	OOV	all	OOV	all	OOV
Arabic	10.3	**98.53**	**94.01**	97.82#	87.44#	98.49#	93.67#	98.52	93.91
English	8.0	93.43	75.39	93.09#	72.03#	**93.82***	**78.67***	93.61*	76.75
French	5.3	95.47	83.29	95.17#	78.30#	95.68*	86.28*	**95.73***	**86.78***
German	11.5	**91.92**	85.63	90.88#	77.70#	91.84	85.32	**91.92**	**85.68**
Hindi	4.4	95.35	76.41	95.07#	71.27#	95.41	77.57	**95.44***	**78.00***
Spanish	6.9	94.82	79.62	95.00	81.17	95.45*	**86.36***	**95.49***	85.84*
Swedish	14.3	95.34	89.80	94.78#	86.04 #	95.57*	90.88*	**95.82***	**92.40***

Table 2: Test set overall OOV rates, POS accuracy in % for baseline, suffix-only baseline, DT similarity and suffix replacement strategies for seven languages.

Unsurprisingly, we observe consistent performance drops, mostly significant, for the *suffix-only* baseline. For all languages except German, the *DT*-based replacement strategies result in significant improvements of either overall accuracy, OOV accuracy or both. In most experiments, the *DT suffix* replacement strategy scores slightly higher than the *DT sim* strategy.

Table 3 lists POS accuracies for three languages for similarities from the *w2v* neural language model in its *SKG* and *CBOW* flavors using the cosine similarity. In contrast to the *DT* -based replacements, there are no improvements over the baseline, and some performance drops are even significant. Also replacing the cosine similarity with the Euclidian distance did not change this

[4] http://corporafromtheweb.org/

[5] http://spraakbanken.gu.se

[6] Using Malt Optimizer (Ballesteros and Nivre, 2016) for the Malt parser; for Mate-tools, we tuned the parameter that represents the percentage of non-projective edges in a language, which matches the parameters suggested by Bohnet (2010).

[7] http://universaldependencies.org/

LANG	SKG				CBOW			
	sim		suffix		sim		suffix	
	all	OOV	all	OOV	all	OOV	all	OOV
Arabic	98.46#	93.39#	98.50#	93.73#	98.48#	93.60#	98.52	93.94
English	93.10#	72.29#	93.57	76.31	93.24#	73.91	93.52	75.70
German	90.99#	77.65#	91.62#	83.61#	91.78	83.92#	91.91	85.43

Table 3: Test set POS accuracies for *w2v*-based model's similarity and suffix replacement strategies for three languages.

observation. The suffix-based strategy seems to work better than the similarity-based strategy also for the *w2v*-based replacement.

It seems that count-based similarities perform better for the replacement. Thus, we did not extend the experiments with *w2v* to other languages.

5.2 Results for Dependency Parsing

As a general trend for all languages (see Table 4), we observe that the graph-based parser achieves higher LAS scores than the transition-based parser.

However, the optimal replacement strategy depends on the language for both parsers. Only for Swedish (*reTAG DT suffix*) and Spanish (*reTAG DT sim*), the same replacements yield the highest scores both on all words and OOV words for both parsers. Using the modified POS tags (*reTAG*) results in improvements for the transitions-based parser for 4 languages and for 5 languages using the graph-based parser. Whereas the results improve only marginal when using the *reTAG* strategy as can be observed from Table 4, most improvements are significant.

Using word embeddings for the *reTAG* strategy (see Table 5), we again observe performance drops, except for Arabic.

Following the *oTAG* strategy, we observe significant improvements on German and Arabic for the CBOW method. For German the best performance is obtained with the SKG model (74.47*) which is slightly higher then the *suffix only* replacement, which achieves high scores in the *oTAG* setting. Whereas for POS tagging the suffix-based DT replacement mostly results in the highest scores, there is no clear recommendation for a replacement strategy for parsing all languages. Looking at the average delta (Δ) values for all languages (see Tables 4 and 5) in comparison to the baseline, the picture is clearer: here, for both parser the *reTAG DT suffix* strategy yields the highest improvements and the CBOW and SKG methods only result in consistent improvements for the oTAG strategy. Further average performance gains are observed for the CBOW suffix-based method using the reTAG strategy.

To sum up, we have noted that the *DT*-based strategies seem more advantageous than the *w2v*-strategies across languages. Comparing the different strategies for using *DTs*, we observe an advantage of *reTAG* over *oTAG* and a slight advantage over *suffix* vs. *sim*. Most notably, *DT reTAG suffix* is the only strategy that never resulted in a significant performance drop on all datasets for both parsers and yields the highest average Δ improvement of 1.50. Given its winning performance on the POS evaluation, we recommend to use this strategy.

6 Data Analysis

6.1 Analysis of POS Accuracy

Since POS quality has a direct influence on parser accuracy, we have analyzed the two *reTag* strategies *suffix* and *sim* for our three similarity sources (*DT*, *SKG*, *CBOW*) in more detail for German and English by comparing them to the *oTAG* baselines. In general, differences are mostly found for open word classes such as ADJ, ADV, NOUN, PROPN and VERB, which naturally have the highest OOV rates in the test data. In both languages, the DT-based strategies supply about 84% of the replacements of the *w2v* strategies.

For German, only the *DT suffix*-based replacements led to a slight overall POS improvement. All similarity sources improved the tagging of NOUN for *suffix*, but not for *sim*. All replacements led to some losses in VERBs, with *SKG* losing the most. Both *w2v* sources lost more on ADJ than the *DT*, which also showed the largest improvements on ADV. In addition, we analyzed the POS classification only for tokens that could be replaced both by the *DT* and the *w2v*-methods. For these tokens, the *SKG* method can not surpass the *oTAG* performance. Furthermore, for *DT* and *CBOW*, the *suffix* strategies achieve slightly lower scores than *sim* (0.18%-0.63%). On the tokens where all methods propose replacements, the *DT* results in better accuracy (86.00%) than *CBOW* (85.82%).

For English, the picture is similar but in general the improvement of the scores is larger: while the *DT sim* led to the largest and the *DT suffix* to the second-largest overall improvements, the *suffix*-based *w2v*-strategies can also improve POS

			oTAG						reTAG					
	baseline		suffix only		DT sim		DT suffix		suffix only		DT sim		DT suffix	
Language	all	OOV	all	OOV	all	OOV	all	OOV	all	OOV	all	OOV	all	OOV
	Graph-based Parser													
Arabic	75.60	56.90	75.61	57.76*	75.74*	58.18*	75.71*	58.31*	74.54#	52.84#	**75.75***	58.18*	75.72*	58.31*
English	79.57	63.64	79.55	63.77	79.64	64.38*	79.54	64.20	79.24#	62.37	**79.95***	**66.17***	79.78*	65.30*
French	77.76	64.59	77.91	65.34	77.61	64.09	77.79	64.84	77.59	64.59	77.59	64.09	**77.97**	**65.84**
German	74.24	68.93	74.43*	**69.66***	74.27	69.14	74.21	69.24	72.26#	63.43#	74.13	68.10	74.22	69.09
Hindi	87.67	72.00	87.76*	72.74	**87.78***	72.80*	87.71	**72.86***	87.49#	70.60	87.67	72.62	87.69	72.74
Spanish	80.02	63.56	80.07	65.28*	80.32*	67.18*	80.30*	66.84*	79.38#	64.59	**80.41***	**68.91***	80.27	68.05*
Swedish	77.13	70.70	77.16	70.87	77.44*	71.07	77.31*	71.03	76.55#	69.12#	77.62*	71.96*	**77.65***	**72.05***
Δ all	0.00	0.00	0.10	0.72	0.10	0.89	0.08	0.93	-0.79	-1.89	0.02	0.95	**0.12**	**1.35**
	Transition-based Parser													
Arabic	72.63	52.81	72.71	53.67	72.79*	53.94*	72.75*	53.91*	71.75#	48.61#	72.77*	53.84*	72.74*	53.84*
English	77.26	61.84	77.15#	61.67	77.16	61.84	77.30	62.41	76.85#	60.14#	77.32	62.33	**77.53***	**63.29***
French	74.25	63.09	74.37	63.84	74.38	64.09	74.24	62.84	74.14	62.34	74.59*	**64.59**	**74.69***	64.09
German	70.29	63.02	70.24	62.97	70.22	62.76	70.29	63.07	67.97#	56.38#	70.21	62.19	70.16	62.34
Hindi	84.08	66.14	83.99#	65.16	**84.16***	**67.24***	84.14*	67.05*	83.78#	63.08#	84.10	66.99	84.14	66.99
Spanish	75.39	57.86	75.52	59.59*	75.67*	59.93*	75.38	59.07	75.19	60.10	**76.10***	**63.90***	75.68	62.52*
Swedish	73.45	66.59	73.48	66.46	73.52	66.66	73.60*	67.02	72.91#	64.61#	74.01*	68.27*	**74.09***	**68.53***
Δ all	0.00	0.00	0.02	0.36	0.11	0.70	0.02	0.53	-0.76	-2.10	0.12	1.01	**0.20**	**1.50**

Table 4: LAS scores for the parsing performance on the test sets when replacing OOV words with a DT. Additionally, we present Δ values for all languages.

	oTAG								reTAG							
	similarity				suffix				similarity				suffix			
	SKG		CBOW		SKG		CBOW		SKG		CBOW		SKG		CBOW	
Language	all	OOV	all	OOV	all	OOV	all	OOV	all	OOV	all	OOV	all	OOV	all	OOV
	Graph-based Parser															
Arabic	75.62	58.00*	75.71*	57.97*	75.67	**58.62***	75.73*	58.49*	75.54	57.66*	75.69	57.83*	75.65	58.42*	75.73*	58.49*
English	79.55	63.85	79.57	64.16	79.58	63.99	79.61	64.03	78.86#	59.97#	79.64	64.12	79.38	62.81	79.57	64.03
German	**74.47***	69.55*	74.39	69.29	74.39*	69.35	74.40*	69.24	72.82#	64.26#	73.70#	66.60#	74.06	67.95	74.14	68.41
Δ all	0.08	0.64	0.08	0.83	0.09	0.65	0.11	0.76	-0.73	-2.53	-0.11	-0.10	-0.13	-0.31	0.01	0.49
	Transition-based Parser															
Arabic	72.62	53.67*	72.65	53.60*	**72.88***	**54.80***	72.72	53.67*	72.60	53.46	72.64	53.49*	72.85*	54.53*	72.71	53.63*
English	77.10#	61.49	77.24	62.06	77.17	62.28	77.28	62.46*	76.54#	57.78#	77.22	61.84	77.07	60.58	77.24	62.37
German	70.19	63.07	70.22	63.38	70.17	**63.54**	**70.36**	63.49	68.90#	57.62#	69.48#	60.68#	69.98#	62.09	70.06	62.60
Δ all	-0.09	0.19	0.01	0.98	-0.02	0.46	0.06	0.65	-0.71	-2.94	-0.09	-0.16	-0.28	-0.55	0.06	0.31

Table 5: LAS scores for the parsing performance replacing OOV words with w2v and Δ values.

tagging quality, whereas the *sim w2v*-strategies decrease POS accuracy. Here, we see improvements for ADJ for all but the *sim*-based *w2v*-strategies, improvements on NOUN for all but *SKG suffix*, and for all *suffix* strategies for VERB. Inspecting again the words that can be replaced by all replacement strategies we observe the highest accuracy improvement using the *suffix* strategies: here the scores outperform the baseline (78.07%) up to 84.00% using the *DT* and up to 80.90% with *CBOW*.

The largest difference and the decisive factor for English and German happens on the PROPN tag: Whereas *DT sim* and *SKG suffix* only result in small positive changes, all other strategies frequently mis-tag PROPN as NOUN, increasing this error class by a relative 15% – 45%. These are mostly replacements of rare proper names with rare nouns, which are less found in *DT* replacements due to the similarity threshold. Regarding the other languages, we found largest improvements in French for NOUN for the *DT sim* replacement, coupled with losses on PROPN. Both *DT* strategies improved VERB. For Spanish largest improvements were found in ADJ, NOUN and PRON for both *DT* strategies. Small but significant improvements for Hindi were distributed across parts of speech, and for Arabic, no sizeable improvements were observed.

Only for Arabic we observe a general performance drop when replacing OOV words. Inspecting the OOV words, we detect that around 97% of these words have been annotated as X (other). Overall, the test set contains 8.4% of such annotations, whereas X is rarely encountered in our other languages. Since the baseline performance for Arabic POS is very high, there is not much to improve with replacements.

6.2 Analysis of Parsing Accuracy by Relation Label

We have conducted a differential analysis comparing LAS F-scores on all our languages between the baseline and the different replacement options, specifically for understanding the effects of *DT reTAG* strategies. Focusing on frequent dependency labels (average occurrence: 4% – 14%), we gain improvements for the relations `conj`, `amod` and `case` across all test sets. Except for Hindi, the LAS F1 score increases up to 0.6% F1 for `case` relations, which is the relation between preposition (or post-positions) and the head noun of the prepositional phrase. For the `amod` relation that connects modifying adjectives to nouns, we observe a +0.5% – +1% improvement in F-score for all languages except Hindi and French, corresponding largely to the increased POS accuracy for nouns and adjectives.

For English, we found most improvements in the relations `compound` (about +1 F1) and `name` (+0.5 – +5.0 F1) for both parsers, while relations `cop` and `xcomp` were recognized less precisely (-0.2 – -0.9 F1). The graph-based parser also improves largely in `appos` (+3.5 – +4.2 F1) and `nmod:npmod` (+5.2 – +6.5 F1), while the transition-based parser sees improvements in `iobj` (+3.8 – +5.1 F1) and `neg` (+1.0 F1). For German, the `case` relation improves for both parsers with +0.2 – +0.6 F1. The graph-based parser improves on `auxpass` (+1.1 – 1.4 F1) and `conj` (+0.4 – +0.9 F1). Whereas pinpointing systematic differences between the two parsers is hardly possible, we often observe that the graph-based parser seems to perform better on rare relations, whereas the transition-based parser deals better with frequent relations.

As with the overall evaluation, there is no clear trend for the *suffix* vs. the *sim* strategy for single relations, except for graph-based German `dobj` and `iobj`, which stayed the same or performed worse for the *DT suffix reTAG* (0 – -0.9 F1), but improved greatly for *DT sim reTAG* (+0.9 – +2.4 F1).

In summary, OOV replacement seems to benefit dependency parsing mostly on relations that involve open class words, as well as relations that need semantic information for disambiguation, e.g. `case`, `dobj` and `iobj`.

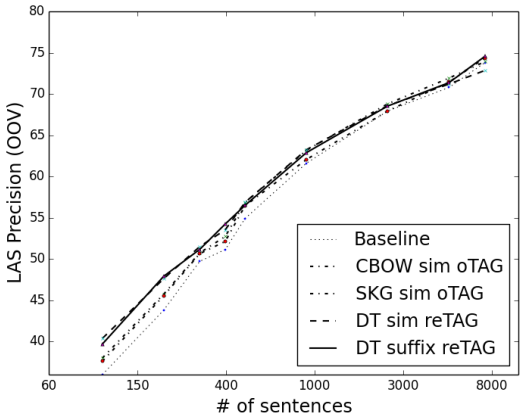

Figure 1: Learning curve of LAS for OOV words for English development set.

7 Discussion

In the following we want to discuss about selecting a recommendation for the OOV replacement and will highlight the differences we observed in our experiments between graph-based and dense-vector-based similarities.

7.1 Recommendations for OOV Replacement

Our experiments show that a simple OOV replacement strategy can lead to significant improvements for dependency parsing across typologically different languages. Improvements can be partially attributed to gains in the POS tagging quality especially with the *suffix*-based replacement strategy, and partially attributed to improved use of lexicalized information from semantic similarity.

Overall, the strategy of replacing OOV words first and POS-tagging the sequence on the basis of the replacements (*reTAG*) shows to be more effective than the other way around. While improvements are generally small yet significant, we still believe that OOV replacement is a viable strategy, especially given its simplicity. In learning curve experiments, as exemplified in Figure 1, we found the relative effect to be more pronounced for smaller amounts of training, despite having less in-vocabulary material to choose from. Thus, our approach seems especially suited for low-resource languages where labeled training material is notoriously scarce.

The question whether to use *DT suffix* or *DT sim* as replacement strategy for dependency parsing is not easily answered – while *DT suffix* shows the best overall improvements across the datasets, *DT*

sim performs slightly better on Arabic and English graph-based parsing and English POS tagging.

7.2 On Differences between Graph-Based and Dense-Vector Similarity

What would be needed to fruitfully utilize the popular neural language model *w2v* as a similarity source, and why does the graph-based *DT* seems to be so much more suited for OOV replacement? From above analysis and from data inspection, we attribute the advantage of *DT* to its capability of NOT returning replacements when it has too low confidence, i.e. no in-vocabulary word is found with a similarity score of 5 or more. In contrast, vector spaces do not provide an interpretable notion of similarity/closeness that can be uniformly applied as a similarity threshold: we have compared cosine similarities of token replacements that lead to improvements, no changes and drops, and found no differences between their average values. A further difference is the structure of the vector space and the *DT* similarity rankings: Whereas the *DT* returns similar words with a frequency bias, i.e. rather frequent words are found in the most similar words per OOV target, the vector space does not have such frequency bias and, since there are more rare than frequent words in language, returns many rare words from the background corpus[8]. This effect can be alleviated to some extent when applying frequency thresholds, but is in turn aggravated when scaling up the background corpus. Thus, a condition that would only take the top-N most similar words from the background collection into account for expansions is also bound to fail for *w2v*. The only reasonable mechanism seems to be a background corpus frequency threshold on the in-vocabulary word. However, even when comparing only on the positions where both *DT* and *w2v* returned replacements, we still find *DT* replacements more advantageous. Inspection revealed that while many replacements are the same for the similarity sources, the *DT* replacements more often stay in the same word class (cf. Table 1), e.g. regarding conjugative forms of verbs and regarding the distinction between common and proper nouns.

[8] we have seen this effect repeatedly and consistently across corpora, languages and parameters

8 Conclusion

In this paper, we have shown that syntactic preprocessing, both POS tagging and dependency parsing, can benefit from OOV replacement. We have devised a simple yet effective strategy (*DT suffix reTAG*) to improve the quality of universal dependency parsing by replacing OOV words via semantically similar words that share a suffix, subsequently run the POS tagger and the dependency parser over the altered sequence, and projecting the labels back to the original sequence. In these experiments similar words from a count-based distributional thesaurus are more effective than the dense numeric `w2v` approach.

In future work, we will apply our method for other types of lexicalized parsers, such as constituency grammar and combinatory categorial grammar parsers, as well as examine the influence of OOVs on semantic tasks like semantic role labeling or frame-semantic parsing.

References

Jacob Andreas and Dan Klein. 2014. How much do word embeddings encode about syntax? In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 822–827, Baltimore, Maryland.

Miguel Ballesteros and Joakim Nivre. 2016. Maltoptimizer: Fast and effective parser optimization. *Natural Language Engineering*, 22:187–213, 3.

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring Continuous Word Representations for Dependency Parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, ACL '14, pages 809–815, Baltimore, MA, USA.

Chris Biemann and Martin Riedl. 2013. Text: Now in 2D! A Framework for Lexical Expansion with Contextual Similarity. *Journal of Language Modelling*, 1(1):55–95.

Bernd Bohnet. 2010. Very High Accuracy and Fast Dependency Parsing is Not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 89–97, Beijing, China.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based N-gram Models of Natural Language. *Computational Linguistics*, 18(4):467–479.

Wenliang Chen, Yue Zhang, and Min Zhang. 2014. Feature Embedding for Dependency Parsing. In *Proceedings of the 25th International Conference on*

Computational Linguistics, COLING 2014, pages 816–826, Dublin, Ireland.

Stefan Evert. 2005. *The Statistics of Word Cooccurrences: Word Pairs and Collocations.* Ph.D. thesis, Institut für maschinelle Sprachverarbeitung, University of Stuttgart.

Rashmi Gangadharaiah, Ralf D. Brown, and Jaime Carbonell. 2010. Monolingual Distributional Profiles for Word Substitution in Machine Translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING 2010, pages 320–328, Beijing, China.

Zellig Sabbetai Harris. 1951. *Methods in Structural Linguistics*. University of Chicago Press, Chicago.

Hen-Hsen Huang, Huan-Yuan Chen, Chang-Sheng Yu, Hsin-Hsi Chen, Po-Ching Lee, and Chun-Hsun Chen. 2014. Sentence Rephrasing for Parsing Sentences with OOV Words. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 26–31, Reykjavik, Iceland.

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-Functional Grammar: A Formal System for Grammatical Representation. In *The Mental Representation of Grammatical Relations*, pages 173–281. MIT Press, Cambridge, MA.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple Semi-supervised Dependency Parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, ACL '08, pages 595–603, Columbus, OH, USA.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Machine Learning*, ICLR 2013, pages 1310–1318, Scottsdale, AZ, USA.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia.

Joakim Nivre. 2009. Non-projective dependency parsing in expected linear time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, ACL '09, pages 351–359, Suntec, Singapore.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. *English Gigaword Fifth Edition*. Linguistic Data Consortium, Philadelphia.

Matthias Richter, Uwe Quasthoff, Erla Hallsteinsdóttir, and Chris Biemann. 2006. Exploiting the Leipzig Corpora Collection. In *Proceedings of the IS-LTC 2006*, pages 68–73, Ljubljana, Slovenia.

Roland Schäfer and Felix Bildhauer. 2013. *Web Corpus Construction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.

Richard Socher, John Bauer, Christopher D. Manning, and Andrew Y. Ng. 2013. Parsing with Compositional Vector Grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, ACL '13, pages 455–465, Sofia, Bulgaria.

Qin Iris Wang, Dale Schuurmans, and Dekang Lin. 2005. Strictly Lexical Dependency Parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology*, Parsing '05, pages 152–159, Vancouver, BC, Canada.

Jiajun Zhang, Feifei Zhai, and Chengqing Zong. 2012. Handling Unknown Words in Statistical Machine Translation from a New Perspective. In *Proceedings of the 1st Conference on Natural Language Processing and Chinese Computing*, NLP&CC '12, pages 176–187, Beijing, China.