From Text to Lexicon: Bridging the Gap between Word Embeddings and Lexical Resources

Ilia Kuznetsov and **Iryna Gurevych**
Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science
Technische Universität Darmstadt
http://www.ukp.tu-darmstadt.de/

Abstract

Distributional word representations (often referred to as word embeddings) are omnipresent in modern NLP. Early work has focused on building representations for word types, and recent studies show that lemmatization and part of speech (POS) disambiguation of targets in isolation improve the performance of word embeddings on a range of downstream tasks. However, the reasons behind these improvements, the qualitative effects of these operations and the combined performance of lemmatized and POS disambiguated targets are less studied. This work aims to close this gap and puts previous findings into a general perspective. We examine the effect of lemmatization and POS typing on word embedding performance in a novel resource-based evaluation scenario, as well as on standard similarity benchmarks. We show that these two operations have complementary qualitative and vocabulary-level effects and are best used in combination. We find that the improvement is more pronounced for verbs and show how lemmatization and POS typing implicitly target some of the verb-specific issues. We claim that the observed improvement is a result of better conceptual alignment between word embeddings and lexical resources, stressing the need for conceptually plausible modeling of word embedding targets.

1 Introduction

Word embeddings learned from unlabeled corpora are one of the cornerstones of modern natural language processing. They offer important advantages over their sparse counterparts, allowing efficient computation and capturing a surprising amount of lexical information about words based solely on usage data.

Since precise word-related terminology is important for this paper, we introduce it early on. A **token** is a unique word occurrence within context. Tokens that are represented by the same sequence of characters belong to the same word **type**. A type signifies one or — in case of morphological ambiguity — several **word forms**. Word forms encode the grammatical information carried by the word and are therefore POS-specific. One of the word forms is considered the **base form** - or **lemma** of the word. All possible word forms of a word represent the **lexeme** of this word. From a semantic perspective, a word is assigned to a minimal sense-bearing **lexical unit** (LU). In general, multi-word and sub-word lexical units are possible, but in this paper we focus on single-word units. LUs are the reference point to the semantics of the word and might be used to describe further properties of the words within a specific **lexicon**, e.g. get assigned one or several senses, related to other LUs via lexical relations etc. Figure 1 illustrates these concepts and positions some of the relevant approaches w.r.t. the level they operate on.

In general, the training objective in the unsupervised word embedding setup (e.g. (Mikolov et al., 2013)) is to induce vector representations for **targets** based on their occurrences in an unlabeled reference corpus. For each occurrence of the target, **context** representation is extracted. The goal is to encode targets so that targets appearing in similar contexts are close in the resulting **vector space model** (VSM). We further refer to the set of targets in a given VSM as **target vocabulary** of this VSM.

The applications of word embeddings can be roughly grouped in two categories, **occurrence-based** and **vocabulary-based**: the former aims to classify tokens (e.g. parsing, tagging, coreference resolution), the

Proceedings of the 27th International Conference on Computational Linguistics, pages 233–244
Santa Fe, New Mexico, USA, August 20-26, 2018.

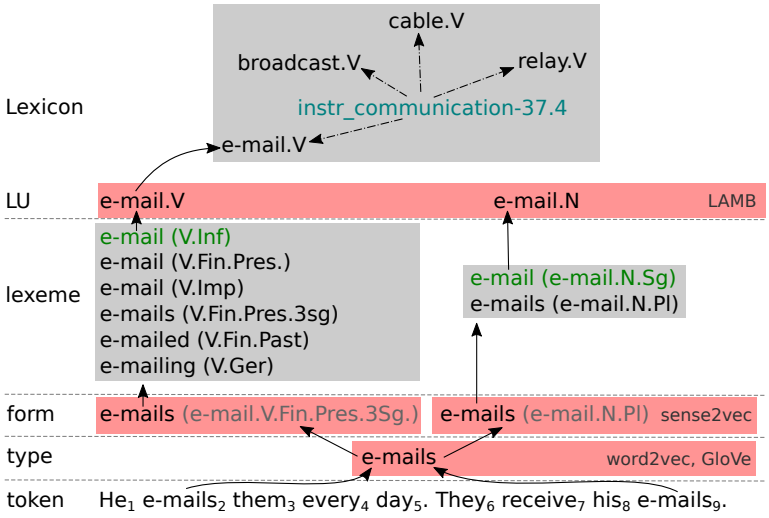

Figure 1: Hierarchy of word-related concepts for *emails*, with VerbNet as lexicon example.

latter aims to classify lexemes (e.g. word clustering, thesaurus construction, lexicon completion). Lexical resources mostly operate on lexeme or lexical unit level. This includes most of the similarity benchmarks (Finkelstein et al., 2001; Bruni et al., 2014; Hill et al., 2015; Gerz et al., 2016) that implicitly provide similarity scores for lexemes and not word types. Traditional word embeddings approaches (Mikolov et al., 2013; Pennington et al., 2014) induce representations on word type level. As our example in Figure 1 shows, this leads to a conceptual gap between the lexicon and the VSM, which has practical consequences. Consider the standard scenario when a type-based VSM is evaluated against a lexeme-based benchmark. One of the types contained in the VSM vocabulary corresponds to the base form (lemma), and the vector for the base word form is used for evaluation. This particular form, however, is selected based on *grammatical* considerations and, as we later demonstrate, is neither the most frequent nor the most representative in terms of contexts, nor the most unambiguous. As a result, (1) the contexts for a given lexical unit are under-represented, ignoring other word forms than the base form; (2) in case of ambiguity the same context-based representation is learned for several different lexemes sharing the same word type.

Recent studies demonstrate that even partially addressing these problems leads to improved performance. Ebert et al. (2016) introduce the lemma-based LAMB embeddings and show that lemmatization of the targets improves the results cross-linguistically on similarity benchmarks and in a WordNet-based evaluation scenario. In our scheme this represents a step up in the conceptual hierarchy but still leaves room for ambiguity on the LU level. Trask et al. (2015) experiment with POS-disambiguated targets, and also report improved performance on a variety of tasks.

While prior work shows that lemmatization and POS-typing of targets in isolation are beneficial for downstream tasks, it does not provide a detailed investigation on why it is the case and does not study the effects of combining the two preprocessing techniques. This paper aims to close this gap. We evaluate the effects of lemmatization and POS disambiguation separately and combined on similarity benchmarks, and further refine our results using a novel resource-based word class suggestion scenario which measures how well a VSM represents VerbNet (Schuler, 2005) and WordNet supersense (Ciaramita and Johnson, 2003) class membership. We find that POS-typing and lemmatization have complementary qualitative and vocabulary-level effects and are best used in combination. We observe that English verb similarity is harder to model and show that using lemmatized and disambiguated embeddings implicitly targets some of the verb-specific issues. In summary, the contributions of this paper are as follows:

- We suggest using lemmatized *and* POS disambiguated targets as a conceptually plausible alternative to type, word form and lemma-based VSMs;
- We introduce the suggestion-based evaluation scenario applicable to a wide range of lexical resources;
- We show that lemmatization and POS disambiguation improve both benchmark and resource-based performance by implicitly targeting some of the grammar-level issues.

2 Related work

The starting point for our study are the results from the lemma-based word embedding approach by Ebert et al. (2016), the POS-disambiguated *sense2vec* embeddings by Trask et al. (2015) and the dependency-based contexts by Levy and Goldberg (2014). Our primary focus is the effect of word embedding targets on vocabulary-based task performance. However, to put our work in context, we provide an extended overview of the issues related to traditional type-based approaches to VSM construction, along with relevant solutions.

Type-based VSMs (Mikolov et al., 2013; Pennington et al., 2014) have several limitations that have been extensively studied. They have **restricted vocabularies** and hence lack the ability to represent out-of-vocabulary words. Several recent approaches treat this issue by learning vector representations for sub-word units, e.g. *fastText* (Bojanowski et al., 2016) and *charagram* (Wieting et al., 2016).

Type-based VSMs do not abstract away from **word inflection**: different forms of the same word are assigned different representations in the VSM. Ebert et al. (2016) introduces lemmatized LAMB embeddings as a way to address this issue and shows that lemmatization is highly beneficial for word similarity and lexical modeling tasks even for morphologically poor English, while using *stems* doesn't lead to significant improvements. Their approach is similar to our `lemma-w2` setup (Section 3). Another recent attempt to address this problem is the study by Vulić et al. (2017b), which uses a small set of morphological rules to specialize the vector space bringing the forms of the same word closer together while setting the derivational antonyms further apart. We relate to this approach in Section 4.

Finally, type-based VSMs neglect the problem of **polysemy**: all senses of a word are encoded as a single entry in the VSM. This issue has been recently approached by multi-modal word embeddings (Athiwaratkun and Wilson, 2017) and Gaussian embeddings (Vilnis and McCallum, 2014). Partially disambiguating the source data via POS tagging has been employed in (Trask et al., 2015). Here, instead of constructing vectors for word forms, POS information is integrated into the vector space as well. This is similar to our `type.POS-w2` setup (Section 3) which, as we show, introduces additional sparsity and ignores morphological inflection.

An alternative line of research aims to learn embeddings for lexical units by using an external WSD tool to preprocess the corpus and applying standard word embedding machinery to induce distributed representations for lexical units, e.g. (Iacobacci et al., 2015; Flekova and Gurevych, 2016). Such approaches require an external WSD tool which introduces additional bias and might not be available for lower-resourced languages. Moreover, to query such VSMs it is necessary to either apply WSD to the input, or to align the inputs with the senses in some other way, which is not always feasible.

From the evaluation perspective, a popular method to assess the performance of a particular vector space model is **similarity benchmarking**. A similarity benchmark consists of word pairs along with human-assessed similarity scores, which are compared to cosine similarities returned by VSMs via rank correlation. Some common examples include WordSim-353 (Finkelstein et al., 2001) and MEN (Bruni et al., 2014). The recent SimLex-999 (Hill et al., 2015) and SimVerb-3500 (Gerz et al., 2016) particularly focus on word similarity as opposed to word relatedness (e.g. *bank* and *money* would be related, but not similar) and have been shown difficult for word embedding models to tackle.

Being attractive from the practical perspective, similarity benchmarks are expensive to create and do not provide detailed insights about the lexical properties encoded by the word embeddings. Motivated by that, **resource-based benchmarking** strategies have been suggested. (Ebert et al., 2016) represents WordNet as a graph and measures the correspondence of similarity ranking to the distances between query words in the graph via mean reciprocal rank. Vulić et al. (2017a) and Sun et al. (2008) apply a clustering algorithm to the input words and measure how well the clusters correspond to the word groupings in VerbNet via purity and collocation. Both methods have certain limitations, which we discuss and attempt to resolve via a novel general suggestion-based approach in Section 5.

3 General setup

Distributional word representations aim to encode word targets based on the context words they co-occur with. One popular general-purpose model for inducing such representations is skip-gram with negative

sampling (SGNS), exemplified by word2vec (Mikolov et al., 2013).

In the original SGNS targets and contexts belong to the same type-based vocabulary, but this is not required by the model. In this study we experiment with the following targets: `type` (*going*), `lemma` (*go*), `type.POS`[1] (*going.V*) and `lemma.POS` (*go.V*). Word embeddings demonstrate qualitative differences depending on context definition (Levy and Goldberg, 2014), and we additionally report the results on 2-word window-based (`w2`) and dependency-based (`dep-W`) contexts. To keep the evaluation setup simple, we only experiment with word type-based contexts.

We train 300-dimensional VSMs on a recent English Wikipedia dump, preprocessed using the Stanford Core NLP pipeline (Manning et al., 2014) with Universal Dependencies 1.0. The text is extracted using the *wikiextractor* module[2] with minor additional cleanup routines. Training the VSM is performed with the skip-gram based *word2vecf* implementation from (Levy and Goldberg, 2014) with default algorithm parameters.

4 Similarity Benchmarks

Following previous work, we first evaluate the effect of lemmatization and POS-typing on similarity benchmarks SimLex-999 (Hill et al., 2015) and SimVerb-3500 (Gerz et al., 2016). Both benchmarks provide POS information, which is required by POS-enriched VSMs. SimLex-999 contains nouns (60%), verbs (30%) and adjectives (10%); SimVerb-3500 is verbs-only. Table 1 summarizes the performance of the VSMs in question on similarity benchmarks.

Context: `w2`

	SimLex				SimVerb
target	N	V	A	all	V
type	.334	**.336**	**.518**	.348	.307
+ POS	.342	.323	.513	.350	.279
lemma	**.362**	.333	.497	**.351**	.400
+ POS	.354	**.336**	.504	.345	**.406**
* type	-	-	-	.339	.277
* type MFit-A	-	-	-	.385	-
* type MFit-AR	-	-	-	.439	.381
Context: `dep-W`					
type	.366	.365	.489	.362	.314
+ POS	.364	.351	.482	.359	.287
lemma	**.391**	.380	**.522**	**.379**	.401
+ POS	.384	**.388**	.480	.366	**.431**
* type	-	-	-	.376	.313
* type MFit-AR	-	-	-	.434	.418

Table 1: Benchmark performance, Spearman's ρ. SGNS results with * taken from (Vulić et al., 2017b). Best results per column (benchmark) annotated for our setup only.

Several observations can be made. Lemmatized targets generally perform better, with the boost being more pronounced on SimVerb. English verbs have richer morphology than other parts of speech and benefit more from lemmatization. Adding POS information benefits the SimVerb and SimLex verb performance, which can be attributed to the **coarse disambiguation** of verb-noun and verb-adjective homonyms. However, the `type.POS` targets show a considerable performance drop on SimVerb and SimLex verbs. This is due to **vocabulary fragmentation**, when the same word type is split into several entries based on the POS, e.g. *acts (110)* $\rightarrow$ *acts.V (80) + acts.N (30)*. As a result, some word types do not exceed the frequency threshold and are not included into the final VSM. Another way to see it

[1] We use coarse POS classes obtained by selecting the first character in the Penn POS tags assigned by the tagger

[2] `https://github.com/attardi/wikiextractor`

is that POS disambiguation increases the sparsity, which lemmatization, in turn, aims to reduce. Using dep-W contexts proves beneficial for both datasets since modeling the context via syntactic dependencies results in more similarity-driven (as opposed to relatedness-driven) word embeddings (Levy and Goldberg, 2014).

We provide the Morph-Fitting scores (Vulić et al., 2017b) as a state-of-the-art reference; a direct comparison is not possible due to the differences in the training data and information available to the models. This approach uses word type-based VSMs specialized via Morph-Fitting (MFit), which can be seen as an alternative to lemmatization. Morph-Fitting consists of two stages: the Attract (A) stage brings word forms of the same word closer in the VSM, while the Repel (R) stage sets the derivational antonyms further apart. Lemma grouping is similar to the Attract stage. However, as the comparison of MFit-A and -AR shows, a major part of the Morph-Fitting performance gain on SimLex comes from the derivational Repel stage[3], which is out of the scope of this paper.

While some properties of lemmatized and POS disambiguated embeddings are visible on the similarity benchmarks, the results are still inconclusive, and we proceed to a more detailed evaluation scenario.

5 Word Class Suggestion

5.1 Background

Similarity benchmarks serve as a standard evaluation tool for word embeddings, but provide little insights into the nature of relationships encoded by the word representations. A more fine-grained context-free evaluation strategy is to assess how well the relationships in a certain **lexical resource** are represented by the given VSM. Two recent approaches to achieve this are rank-based and clustering-based evaluation.

Rank-based evaluation treats the lexical resource as a graph with entries as nodes and lexical relations as edges, and estimates how well the similarities between the VSM targets represent the distances in this graph via mean reciprocal rank (MRR), e.g. Ebert et al. (2016) use WordNet (Miller, 1995). It requires the target lexical resource to have a dense linked structure which might be not present.

Clustering-based evaluation, e.g. Vulić et al. (2017a) utilize the VSM to produce target clusters which are compared to the groupings from the lexical resource via collocation and purity. This approach only requires lexical entries to be grouped into classes. However, it doesn't model word ambiguity: a word can only be assigned to a single cluster. Moreover, it introduces an additional level of complexity (clustering algorithm and its parameters) which might obscure the VSM performance details.

In this paper we propose an alternative, suggestion-based evaluation approach: we use the source VSM directly to generate **word class suggestions** (WCS) for a given input term. Many lexical resources group words into intersecting word classes, providing a compact way to describe word properties on the class level. For example, in VerbNet (Schuler, 2005) verbs can belong to one or more Levin classes (Levin, 1993) based on their syntactic behavior, FrameNet (Baker et al., 1998) groups its entries by the semantic frames they can evoke, and WordNet (Miller, 1995) provides coarse-grained supersense groupings. Suggestion-based evaluation can be seen as flexible alternative to clustering-based evaluation, which intrinsically takes ambiguity into account and does not require an additional clustering layer. The following section describes the suggestion-based evaluation in more detail[4].

5.2 Task formulation

Abstracting away from the resource specifics, a **lexicon** L defines a mapping from a set of **members** $m_1, m_2, ...m_i \in M$ to a set of **word classes** $c_1, c_2...c_j \in C$. We further denote the set of classes available for a member m as $L(m)$ and the set of members for a given class c as $L'(c)$. Given a **query** q, our task is to provide a set of word class suggestions $WCS_L(q) = \{c_a, c_b...\} \in C$. Note that we aim to predict all *potential* classes for a member given its vector representation on vocabulary level, independent of context.

[3] See also (Vulić et al., 2017b), Table 5.

[4] Our implementation is available at https://github.com/UKPLab/coling2018-wcs

5.3 Suggestion strategies

Given an input target w, the source vector space VSM provides its vector representation $VSM(w)$. We use cosine similarity between targets $sim(w_i, w_j)$ to rank the word classes.

A lexical resource might already contain a substantial number of members, and a natural solution for word class suggestion is to find the **prototype** member m_{proto} closest to the query q in the VSM, and use its classes as suggestions. This scenario mimics human interaction with the lexicon. If $q \in M$, this is equivalent to a lexicon lookup. More formally,

$$m_{proto} = argmax_{m \in M} sim(q, m) \qquad score_{proto}(q, c, L) = \begin{cases} 1, & \text{if } c \in L(m_{proto}) \\ 0, & \text{otherwise} \end{cases}$$

The prototype strategy per se is sensitive to the coverage gaps in the lexicon and inconsistencies in the VSM. We generalize it by ranking *each* word class $c \in C$ using the similarity between the query q and its closest member in c:

$$score_{top}(q, c, L) = max_{m \in L'(c)} sim(q, m)$$

This is equivalent to performing the prototype search on each word class, and scoring each class by the closest prototype among its members, given the query. We use this generalized strategy for our WCS experiments throughout this paper. The output of the model $WCS_L(q, VSM)$ is a set of classes ranked using the input VSM, as illustrated by Table 2.

query	top classes / prototypes	query	top classes / prototypes
dog→	animal (*cat*), food (*rabbit*) ...	idea→	cognition (*concept*), artifact (*notion*) ...
crane→	artifact (*derrick*), animal (*skimmer*) ...	bug→	animal (*worm*), state (*flaw*) ...

Table 2: WCS output for WordNet supersenses

5.4 Evaluation procedure

For each member m in the lexicon in turn, we remove it from the lexicon, resulting in a reduced lexicon L_{-m}. We aim to reconstruct its classes using the suggestion algorithm and the remaining mappings.

The performance is measured via precision (P) and recall (R) at rank k with the list of original classes for a given lexical unit serving as *gold* reference. Formally, given m we compute the ranking $WCS_{L_{-m}}(m, VSM)$. Let S_k be the set of classes suggested by the system up to the rank k, and T be the true set of classes for a given member in the original lexicon. Then

$$P_{@k} = \frac{|S_k \cap T|}{|S_k|} \qquad R_{@k} = \frac{|S_k \cap T|}{|T|}$$

To get a single score, we average individual members' $P_{@k}$ and $R_{@k}$ for each value of k, resulting in scores $\overline{P}_{@k}$ and $\overline{R}_{@k}$. F-measure might be then calculated using the standard formula

$$F_{@k} = \frac{2\overline{P}_{@k}\overline{R}_{@k}}{\overline{P}_{@k} + \overline{R}_{@k}}$$

Upper bound Since the number of gold classes is not known in advance, the evaluation is always performed on k ranks, which leads to a resource-specific upper bound on P. For example, if a member only has one class, the ranked list of 10 suggestions will inevitably show lower precision. When the member set is not fully covered by the VSM target vocabulary, additional upper bound on R applies.

6 Experiment

6.1 Resources

We use two lexical resources for suggestion-based evaluation: VerbNet 3.3 (Schuler, 2005) and WordNet 3.1 (Miller, 1995). VerbNet groups verbs into classes[5] so that verbs in the same class share syntactic behavior, predicate semantics, semantic roles and restrictions imposed on these roles. For example, the verb *buy* belongs to the class *get-13.5.1*. The class specifies a set of available roles, e.g. an animate `Agent` (buyer), an `Asset` (price paid) and a `Theme` (thing bought), and lists available syntactic constructions, e.g. the `Asset V Theme` construction (*"$50 won't buy a dress"*). A verb might appear in several classes, indicating different verb senses. For example, the verb *hit* allows several readings: as *hurt* (*John hit his leg*), as *throw* (*John hit Mary the ball*) and as *bump* (*The cart hit against the wall*). VerbNet has been successfully used to support semantic role labeling (Giuglea and Moschitti, 2006), information extraction (Mausam et al., 2012) and semantic parsing (Shi and Mihalcea, 2005).

WordNet, besides providing a dense network of lexical relations, groups its entries into coarse-grained supersense classes, e.g. `noun.animal` (*aardvark*, *koala*), `noun.location` (*park*, *senegal*), `noun.time` (*forties*, *nanosecond*). WordNet supersense tags have been applied to a range of downstream tasks, e.g. metaphor identification and sentiment polarity classification (Flekova and Gurevych, 2016). WordNet differs from VerbNet in terms of granularity, member-class distribution and part of speech coverage, and allows us to estimate VSM performance on nominal as well as verbal supersenses, which we evaluate separately. Table 3 provides the statistics for the resources. We henceforth denote VerbNet as `VN`, WordNet nominal supersense lexicon as `WN-N` and WordNet verbal supersense lexicon as `WN-V`.

	classes	members	ambig	%ambig
VN	329	4 569	1 366	30%
WN-V	15	8 702	3 326	38%
WN-N	26	57 616	9 907	17%

Table 3: Lexicon statistics, single-word members

6.2 Results

We use the suggestion-based evaluation to examine the effect of lemmatization and POS-disambiguation. We first analyze the coverage of the VSMs in question with respect to the lexica at hand, see Table 4. For brevity we only report coverage on `w2` contexts[6].

target	VN	WN-V	WN-N
type	81	66	47
+POS	54	39	43
lemma	88	76	53
+POS	79	63	50
shared	54	39	41

Table 4: Lexicon member coverage (%)

Coverage analysis on lexica confirms our previous observations: lemmatization allows more targets to exceed the SGNS frequency threshold, which results in consistently better coverage. POS-disambiguation, in turn, fragments the vocabulary and consistently reduces the coverage with the effect being less pronounced for lemmatized targets. `WN-N` shows low coverage containing many low-frequency members. Due to significant discrepancies in VSM coverage, we conduct our experiments on **shared vocabulary**, only including members found in all VSMs to analyze the qualitative differences between VSMs.

[5]For simplicity in this study we ignore subclass divisions.

[6]We have observed slight coverage differences for `dep` contexts, and attribute this to the context vocabulary fragmentation caused by dependency typing of the contexts, similar to the POS fragmentation effect described earlier.

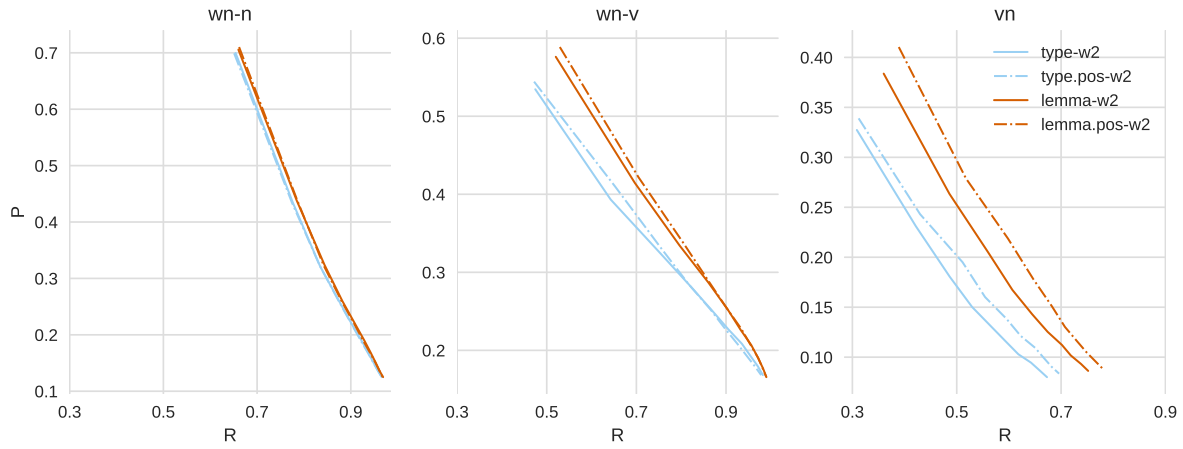

Figure 2: WCS PR-curve, shared vocabulary, `w2` contexts.

We treat the cutoff rank k as a parameter that specifies the Precision-Recall trade-off. As Figure 2 demonstrates, lemmatized targets consistently outperform their word form-based counterparts on the WCS task. The magnitude of improvements varies between resources: verb-based `WN-V` and `VN` benefit more from lemmatization, and `VN` gains most from POS-disambiguation. This aligns with the similarity benchmarking results where the verb-based SimVerb benefits more from addition of lemma and POS information.

	WN-N			WN-V			VN		
	P	R	F	P	R	F	P	R	F
Context: `w2`									
type	.700	.654	.676	.535	.474	.503	.327	.309	.318
+POS	.699	.651	.674	.544	.472	.505	.339	.312	.325
lemma	.706	.660	.682	.576	.520	.547	.384	.360	.371
+POS	**.710**	**.662**	**.685**	**.589**	**.529**	**.557**	**.410**	**.389**	**.399**
Context: `dep`									
type	.712	.661	.686	.545	.457	.497	.324	.296	.310
+POS	.715	.659	.686	.560	.464	.508	.349	.320	.334
lemma	**.725**	**.668**	**.696**	.591	.512	.548	.408	.371	.388
+POS	.722	.666	.693	**.609**	**.527**	**.565**	**.412**	**.381**	**.396**

Table 5: WCS performance, shared vocabulary, $k = 1$. Best results across VSMs in bold.

Table 5 provides exact scores for reference. Note that the shared vocabulary setup puts the `type` and `type.POS` VSMs at advantage since it eliminates the effect of low coverage. Still, `lemma`-based targets significantly[7] ($p \leq .005$) outperform `type`-based targets in terms of F-measure in all cases. For window-based `w2` contexts POS disambiguation yields significantly better F scores on lemmatized targets for `VN` ($p \leq .005$) with borderline significance for `WN-N` and `WN-V` ($p \approx .05$). When dependency-based `dep` contexts are used, the effect of POS disambiguation is only statistically significant on `type` targets for `VN` ($p \leq .005$) and on `lemma`-based targets for `WN-V` ($p \leq .005$). We attribute this to the fact that dependency relations used by `dep` contexts are highly POS-specific, reducing the effect of explicit disambiguation. Lemma-based targets without POS disambiguation perform best on `WN-N` when dependency-based contexts are used; however, the difference to lemmatized *and* disambiguated targets is not statistically significant ($p > .1$).

Our results are in line with the previous observations (Gerz et al., 2016) in that the verb similarity is

[7]Wilcoxon signed-rank test over individual lexicon members' F scores

	SimLex			SimVerb	WN-N	WN-V	VN
	N	V	A	V	N	V	V
base	.80	.26	1.0	.24	.86	.21	.22
avg #POS	1.08	1.01	1.39	1.50	1.15	1.37	1.42
single POS	.93	.99	.62	.51	.85	.65	.59

Table 6: Base form ratio and available POS averaged over members; % members with single POS.

hard to capture with standard VSMs. To investigate why verbs benefit most from lemmatization and POS disambiguation, we analyze some relevant statistics based on our Wikipedia data. Table 6 shows the ratio of base form to total occurrences in our corpus[8]. As we can see, the base form (lemma) is by far not the dominating form for verbs. Practically this means that for our resources verbal `type` targets have direct access to only 20-25% of the corpus occurrence data on average. Individual verbs and nouns also differ in terms of preferred word forms, which introduces additional bias into the evaluation. This effect is countered by lemmatization.

The second important difference between the noun- and verb-based lexica is the number of POS available for the lexicon members' lemmas. As Table 6 further demonstrates, nouns are less ambiguous in terms of POS: for example, `VN` member lemmas appear with 1.42 distinct POS categories on average, compared to 1.15 categories for `WN-N`. Individual members might differ in terms of POS frequency distribution, again biasing the evaluation. One regular phenomenon accountable for this is the *verbification* of nouns and adjectives, when a verbal form is constructed without adding any derivational markers. While these derivations might to some extent preserve the similarities between words (e.g. *e-mail.N→e-mail.V* is similar to *fax.N→fax.V*), many cases are less transparent and benefit from POS separation (e.g. the meaning shift in *air.N→air.V* is different from *water.N→water.V*). One exception is the verb subset of SimLex which has a particularly low POS ambiguity.

7 Future work

Lexical unit-based modeling. In this work we have shown that lemmatization and subsequent POS disambiguation benefit both benchmark- and resource-based performance of word embeddings. While verb semantics is notoriously hard to pinpoint per se, we show that modeling verbs via type-based distributed representations also meets grammar-related challenges which can be partially addressed with lemmatization and POS-disambiguation of the inputs. From the conceptual perspective, lemmatized and POS-tagged targets can be seen as another step towards conceptually plausible lexical unit-based modeling of word usage. In this work we focused on single-word entities, and analyzing the effect of including multi-word expressions into the VSM vocabulary is an important direction for future studies.

Word embedding methods. To ensure fair comparison and to keep our evaluation setup compact, we have consciously restricted the scope of the study to a single word embedding model (SGNS), single context size (`w2`) and a single parameter set (*word2vecf* SGNS default). Our results could be further validated by experimenting with alternative context definitions and word embedding models, e.g. GloVe (Pennington et al., 2014) and CBOW (Mikolov et al., 2013). Experiments on character-based models, e.g. *fasttext* (Bojanowski et al., 2016) or *charagram* (Wieting et al., 2016) could be another interesting extension to our work. However, it is not clear how to integrate lemma and POS information with character-based representations in an elegant way.

Cross-linguistic studies. POS tagging and lemmatization are general and well-defined language-independent operations. We have focused on English and have shown that POS-typing and lemmatization implicitly target several grammar-level issues in the context of word embeddings. (Ebert et al., 2016) demonstrate that the improvements from lemmatization hold in a cross-lingual setup. While we believe our results to generally hold cross-linguistically, the relative contribution of POS disambiguation and

[8]We use all lemmas that appear more than 100 times in the corpus; to smooth the effect of tagging errors we only count POS that appear with the target lemma in more than 10% of total lemma occurrences. All statistics averaged over individual lemmas.

lemmatization will inevitably depend on the grammatical properties of the language, constituting another topic for further research.

Suggestion-based evaluation. Our evaluation procedure has clear advantages: it is word class-based, polysemy-aware, does not introduce additional complexity and does not require an annotated corpus. It is resource-agnostic and only requires the target lexical resource to group words into classes. However, several issues must be addressed before it can be used on large scale. Our leave-one-out scenario excludes *singleton classes*, i.e. classes that only have one member. This issue will become less severe with resource coverage increasing over time. The evaluation depends on resource and VSM vocabulary coverage, and for qualitative comparison between VSMs vocabulary intersection should always be taken into account. Alternative suggestion strategies might be explored, e.g. averaging among class members instead of selecting the closest prototype.

Application scenarios. We have introduced word class suggestion as an evaluation benchmark for word embeddings. However, the WCS output might be used in vocabulary-based **application scenarios**, e.g. as annotation study support in cases when the lexicon is available, but the usage corpora are scarce; as a lexicographer tool for finding the gaps in existing lexica; and as a source for context-independent unknown word class candidates in a word sense disambiguation setup.

8 Conclusion

This paper has investigated the effect of lemmatized and POS-disambiguated targets on vector space model performance. We have shown that these preprocessing operations lead to improved performance on the standard benchmarks as well as in a novel word class suggestion scenario. Our results stress the need for more conceptually sound distributional models which align better to lexical resources. This would help to abstract away from the grammar-level issues and allow to further focus on bridging the gap between distributional models and lexical resources on the semantic level.

Acknowledgements

This work has been supported by the German Research Foundation as part of the QA-EduInf project (grant GU 798/18-1 and grant RI 803/12-1) and by the FAZIT Stiftung. We would like to thank the anonymous reviewers for useful comments and future research suggestions.

References

Ben Athiwaratkun and Andrew Wilson. 2017. Multimodal word distributions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1645–1656, Vancouver, Canada, July. ACL.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*, pages 86–90, Stroudsburg, PA, USA. ACL.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49(1):1–47, January.

Massimiliano Ciaramita and Mark Johnson. 2003. Supersense tagging of unknown nouns in wordnet. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 168–175, Stroudsburg, PA, USA. ACL.

Sebastian Ebert, Thomas Müller, and Hinrich Schütze. 2016. Lamb: A good shepherd of morphologically rich languages. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, USA, November. ACL.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM.

Lucie Flekova and Iryna Gurevych. 2016. Supersense embeddings: A unified model for supersense interpretation, prediction and utilization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, volume 1, pages 2029–2041. ACL.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. Simverb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182. ACL.

Ana-Maria Giuglea and Alessandro Moschitti. 2006. Semantic role labeling via framenet, verbnet and propbank. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 929–936, Stroudsburg, PA, USA. ACL.

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

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015*, pages 95–105. ACL.

Beth Levin. 1993. *English verb classes and alternations : a preliminary investigation*. University of Chicago Press, Chicago, IL.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, June. ACL.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60. ACL.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534, Stroudsburg, PA, USA. ACL.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

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

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

Karin Kipper Schuler. 2005. *Verbnet: A Broad-coverage, Comprehensive Verb Lexicon*. Ph.D. thesis, University of Pennsylvania, Philadelphia, PA, USA.

Lei Shi and Rada Mihalcea. 2005. Putting pieces together: Combining framenet, verbnet and wordnet for robust semantic parsing. In *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'05, pages 100–111, Mexico City, Mexico. Springer-Verlag.

Lin Sun, Anna Korhonen, and Yuval Krymolowski. 2008. Verb class discovery from rich syntactic data. In *Proceedings of the 9th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'08, pages 16–27, Haifa, Israel. Springer-Verlag.

Andrew Trask, Phil Michalak, and John Liu. 2015. sense2vec-a fast and accurate method for word sense disambiguation in neural word embeddings. *arXiv preprint arXiv:1511.06388*.

Luke Vilnis and Andrew McCallum. 2014. Word representations via gaussian embedding. *arXiv preprint arXiv:1412.6623*.

Ivan Vulić, Nikola Mrkšić, and Anna Korhonen. 2017a. Cross-lingual induction and transfer of verb classes based on word vector space specialisation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2546–2558, Copenhagen, Denmark, September. ACL.

Ivan Vulić, Nikola Mrkšić, Roi Reichart, Diarmuid Ó Séaghdha, Steve Young, and Anna Korhonen. 2017b. Morph-fitting: Fine-tuning word vector spaces with simple language-specific rules. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 56–68, Vancouver, Canada, July. ACL.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Charagram: Embedding words and sentences via character n-grams. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1515, Austin, Texas, US, November. ACL.