Morph-fitting: Fine-Tuning Word Vector Spaces with Simple Language-Specific Rules

Ivan Vulić1, Nikola Mrkšić1, Roi Reichart2
Diarmuid Ó Séaghdha3, Steve Young1, Anna Korhonen1

1 University of Cambridge 2 Technion, Israel Institute of Technology 3 Apple Inc.
{iv250,nm480,sjy11,alk23}@cam.ac.uk
doseaghdha@apple.com roiri@ie.technion.ac.il

Abstract

Morphologically rich languages accentuate two properties of distributional vector space models: 1) the difficulty of inducing accurate representations for low-frequency word forms; and 2) insensitivity to distinct lexical relations that have similar distributional signatures. These effects are detrimental for language understanding systems, which may infer that inexpensive is a rephrasing for expensive or may not associate acquire with acquires. In this work, we propose a novel morph-fitting procedure which moves past the use of curated semantic lexicons for improving distributional vector spaces. Instead, our method injects morphological constraints generated using simple language-specific rules, pulling inflectional forms of the same word close together and pushing derivational antonyms far apart. In intrinsic evaluation over four languages, we show that our approach: 1) improves low-frequency word estimates; and 2) boosts the semantic quality of the entire word vector collection. Finally, we show that morph-fitted vectors yield large gains in the downstream task of dialogue state tracking, highlighting the importance of morphology for tackling long-tail phenomena in language understanding tasks.

1 Introduction

Word representation learning has become a research area of central importance in natural language processing (NLP), with its usefulness demonstrated across many application areas such as parsing (Chen and Manning, 2014; Johannsen et al., 2015), machine translation (Zou et al., 2013), and many others (Turian et al., 2010; Collobert et al., 2011). Most prominent word representation techniques are grounded in the distributional hypothesis (Harris, 1954), relying on word co-occurrence information in large textual corpora (Curran, 2004; Turney and Pantel, 2010; Mikolov et al., 2013; Mnih and Kavukcuoglu, 2013; Levy and Goldberg, 2014; Schwartz et al., 2015, i.a.).

Morphologically rich languages, in which “substantial grammatical information... is expressed at word level” (Tsarfaty et al., 2010), pose specific challenges for NLP. This is not always considered when techniques are evaluated on languages such as English or Chinese, which do not have rich morphology. In the case of distributional vector space models, morphological complexity brings two challenges to the fore:

1. Estimating Rare Words: A single lemma can have many different surface realisations. Naively treating each realisation as a separate word leads to sparsity problems and a failure to exploit their shared semantics. On the other hand, lemmatising the entire corpus can obfuscate the differences that exist between different word forms even though they share some aspects of meaning.

2. Embedded Semantics: Morphology can encode semantic relations such as antonymy (e.g. literate and illiterate, expensive and inexpensive) or (near-)synonymy (north, northern, northerly).

In this work, we tackle the two challenges jointly by introducing a resource-light vector space fine-tuning procedure termed morph-fitting. The proposed method does not require curated knowledge bases or gold lexicons. Instead, it makes use of the observation that morphology implicitly encodes semantic signals pertaining to synonymy (e.g., German word inflections katalanisch, katalanischer denote the same semantic concept in different grammatical roles), and antonymy (e.g., mature vs. immature), capitalising on the
proliferation of word forms in morphologically rich languages. Formalised as an instance of the post-processing semantic specialisation paradigm (Faruqui et al., 2015; Mrkšić et al., 2016), morph-fitting is steered by a set of linguistic constraints derived from simple language-specific rules which describe (a subset of) morphological processes in a language. The constraints emphasise similarity on one side (e.g., by extracting morphological synonyms), and antonymy on the other (by extracting morphological antonyms), see Fig. 1 and Tab. 2.

The key idea of the fine-tuning process is to pull synonymous examples described by the constraints closer together in the transformed vector space, while at the same time pushing antonymous examples away from each other. The explicit post-hoc injection of morphological constraints enables: a) the estimation of more accurate vectors for low-frequency words which are linked to their high-frequency forms by the constructed constraints; and b) specialising the distributional space to distinguish between similarity and relatedness (Kiela et al., 2015), thus supporting language understanding applications such as dialogue state tracking (DST).2

As a post-processor, morph-fitting allows the integration of morphological rules with any distributional vector space in any language: it treats an input distributional word vector space as a black box and fine-tunes it so that the transformed space reflects the knowledge coded in the input morphological constraints (e.g., Italian words rispettoso and irrispettoso should be far apart in the transformed vector space, see Fig. 1). Tab. 1 illustrates the effects of morph-fitting by qualitative examples in three languages: the vast majority of nearest neighbours are “morphological” synonyms.

We demonstrate the efficacy of morph-fitting in four languages (English, German, Italian, Russian), yielding large and consistent improvements on benchmarking word similarity evaluation sets such as SimLex-999 (Hill et al., 2015), its multilingual extension (Leviant and Reichart, 2015), and SimVerb-3500 (Gerz et al., 2016). The improvements are reported for all four languages, and with a variety of input distributional spaces, verifying the robustness of the approach.

We then show that incorporating morph-fitted vectors into a state-of-the-art neural-network DST model results in improved tracking performance, especially for morphologically rich languages. We report an improvement of 4% on Italian, and 6% on German when using morph-fitted vectors instead of the distributional ones, setting a new state-of-the-art DST performance for the two datasets.3

---

1For instance, the vector for the word katalanischem which occurs only 9 times in the German Wikipedia will be pulled closer to the more reliable vectors for katalanisch and katalanischer, with frequencies of 2097 and 1383 respectively.

2Representation models that do not distinguish between synonyms and antonyms may have grave implications in downstream language understanding applications such as spoken dialogue systems: a user looking for ‘an affordable Chinese restaurant in west Cambridge’ does not want a recommendation for ‘an expensive Thai place in east Oxford’.

3There are no readily available DST datasets for Russian.
2 Morph-fitting: Methodology

Preliminaries In this work, we focus on four languages with varying levels of morphological complexity: English (EN), German (DE), Italian (IT), and Russian (RU). These correspond to languages in the Multilingual SimLex-999 dataset. Vocabularies \( W_{en}, W_{de}, W_{it}, W_{ru} \) are compiled by retaining all word forms from the four Wikipedias with word frequency over 10, see Tab. 3. We then extract sets of linguistic constraints from these (large) vocabularies using a set of simple language-specific if-then-else rules, see Tab. 2. These constraints (Sect. 2.2) are used as input for the vector space post-processing ATTRACT-REPEL algorithm (outlined in Sect. 2.1).

2.1 The ATTRACT-REPEL Model

The ATTRACT-REPEL model, proposed by Mrkšić et al. (2017b), is an extension of the PARAGRAM procedure proposed by Wieting et al. (2015). It provides a generic framework for incorporating similarity (e.g. successful and accomplished) and antonymy constraints (e.g. nimble and clumsy) into pre-trained word vectors. Given the initial vector space and collections of ATTRACT and REPEL constraints \( A \) and \( R \), the model gradually modifies the space to bring the designated word vectors closer together or further apart. The method’s cost function consists of three terms. The first term pulls the ATTRACT examples \((x_l, x_r) \in A \) closer together. If \( B_A \) denotes the current mini-batch of ATTRACT examples, this term can be expressed as:

\[
A(B_A) = \sum_{(x_l, x_r) \in B_A} \left( ReLU \left( \delta_{att} + x_l x_t - x_l x_r \right) \right)
\]

where \( \delta_{att} \) is the similarity margin which determines how much closer synonymous vectors should be to each other than to each of their respective negative examples. \( ReLU(x) = \max(0, x) \) is the standard rectified linear unit (Nair and Hinton, 2010). The ‘negative’ example \( x_t \) for each word \( x_l \) in any ATTRACT pair is the word vector closest to \( x_l \) among the examples in the current mini-batch (distinct from its target synonym and \( x_l \) itself). This means that this term forces synonymous words from the in-batch ATTRACT constraints to be closer to one another than to any other word in the current mini-batch.

The second term pushes antonyms away from each other. If \((x_l, x_r) \in B_R \) is the current mini-batch of REPEL constraints, this term can be expressed as follows:

\[
R(B_R) = \sum_{(x_l, x_r) \in B_R} \left( ReLU \left( \delta_{rep} + x_l x_t - x_l x_r \right) \right)
\]

where \( \delta_{rep} \) is the current mini-batch of REPEL constraints, this term can be expressed as follows:

\[
R(B_A, B_R) = \sum_{x_l \in V(B_A), x_r \in V(B_R)} \lambda_{reg} \| x_l^{init} - x_r \|_2
\]

where \( \lambda_{reg} \) is the L2 regularisation constant. This term effectively pulls word vectors towards their initial (distributional) values, ensuring that relations encoded in initial vectors persist as long as they do not contradict the newly injected ones.

2.2 Language-Specific Rules and Constraints

Semantic Specialisation with Constraints The fine-tuning ATTRACT-REPEL procedure is entirely driven by the input ATTRACT and REPEL sets of constraints.

---

A native speaker can easily come up with these sets of morphological rules (or at least with a reasonable subset of them) without any linguistic training. What is more, the rules for DE, IT, and RU were created by non-native, non-fluent speakers with a limited knowledge of the three languages, exemplifying the simplicity and portability of the approach.

---

Table 2: Example synonymous (infectional; top) and antonymous (derivational; bottom) constraints.

<table>
<thead>
<tr>
<th>English</th>
<th>German</th>
<th>Italian</th>
</tr>
</thead>
<tbody>
<tr>
<td>discuss, discussed</td>
<td>(schottisch, schottischem)</td>
<td>(golfo, golfi)</td>
</tr>
<tr>
<td>laugh, laughing</td>
<td>(dramailge, damalgene)</td>
<td>(minato, minata)</td>
</tr>
<tr>
<td>(pacifist, pacifists)</td>
<td>(kombiniere, kombinierte)</td>
<td>(metterre, metto)</td>
</tr>
<tr>
<td>(vacuate, evacuated)</td>
<td>(schweitg, schweitg)</td>
<td>(creoscono, cresci)</td>
</tr>
<tr>
<td>(evaluate, evaluates)</td>
<td>(hacken, gehackt)</td>
<td>(credeti, credete)</td>
</tr>
<tr>
<td>(dressed, undressed)</td>
<td>(stabil, unstabil)</td>
<td>(abitata, inabitato)</td>
</tr>
<tr>
<td>(similar, dissimilar)</td>
<td>(geformtes, ungeformt)</td>
<td>(realbi, irrealbi)</td>
</tr>
<tr>
<td>(formality, informality)</td>
<td>(relevant, irrelevant)</td>
<td>(attuato, inattuato)</td>
</tr>
</tbody>
</table>

---

5We use hyperparameter values \( \delta_{att} = 0.6, \delta_{rep} = 0.0, \lambda_{reg} = 10^{-5} \) from prior work without fine-tuning. We train all models for 10 epochs with AdaGrad (Duchi et al., 2011).
<table>
<thead>
<tr>
<th>Language</th>
<th>ATTRACT</th>
<th>REPEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>1,368,891</td>
<td>231,448</td>
</tr>
<tr>
<td>German</td>
<td>1,216,161</td>
<td>648,344</td>
</tr>
<tr>
<td>Italian</td>
<td>541,779</td>
<td>278,974</td>
</tr>
<tr>
<td>Russian</td>
<td>950,783</td>
<td>408,400</td>
</tr>
</tbody>
</table>

Table 3: Vocabulary sizes and counts of ATTRACT (A) and REPEL (R) constraints.

These can be extracted from a variety of semantic databases such as WordNet (Fellbaum, 1998), the Paraphrase Database (Ganitkevitch et al., 2013; Pavlick et al., 2015), or BabelNet (Navigli and Ponzetto, 2012; Ehrmann et al., 2014) as done in prior work (Faruqui et al., 2015; Wieting et al., 2015; Mrkšić et al., 2016, i.a.). In this work, we investigate another option: extracting constraints without curated knowledge bases in a spectrum of languages by exploiting inherent language-specific properties related to linguistic morphology. This relaxation ensures a wider portability of ATTRACT-REPEL to languages and domains without readily available or adequate resources.

**Extracting ATTRACT Pairs** The core difference between inflectional and derivational morphology can be summarised in a few lines as follows: the former refers to a set of processes through which the word form expresses meaningful syntactic information, e.g., verb tense, without any change to the semantics of the word. On the other hand, the latter refers to the formation of new words with semantic shifts in meaning (Schone and Jurafsky, 2001; Haspelmath and Sims, 2013; Lazaridou et al., 2013; Zeller et al., 2013; Cotterell and Schütze, 2017).

For the ATTRACT constraints, we focus on inflectional rather than derivational morphology rules as the former preserve the full meaning of a word, modifying it only to reflect grammatical roles such as verb tense or case markers (e.g., (en_read, en_reads)) or ((de_katalanisch, de_katalanischer)). This choice is guided by our intent to fine-tune the original vector space in order to improve the embedded semantic relations.

We define two rules for English, widely recognised as morphologically simple (Avramidis and Koehn, 2008; Cotterell et al., 2016b). These are: (R1) if \( w_1, w_2 \in W_{en} \), where \( w_2 = w_1[: -1] + \text{ing/ed/s} \), then add \((w_1, w_2)\) and \((w_2, w_1)\) to the set of ATTRACT constraints \( A \). This rule yields pairs such as (look, looks), (look, looking), (look, looked).

If \( w[: -1] \) is a function which strips the last character from word \( w \), the second rule is: (R2) if \( w_1 \) ends with the letter \( e \) and \( w_1 \in W_{en} \) and \( w_2 \in W_{en} \), where \( w_2 = w_1[: -1] + \text{ing/ed} \), then add \((w_1, w_2)\) and \((w_2, w_1)\) to \( A \). This creates pairs such as (create, creating) and (create, created). Naturally, introducing more sophisticated rules is possible in order to cover for other special cases and morphological irregularities (e.g., sweep / swept), but in all our EN experiments, \( A \) is based on the two simple EN rules R1 and R2.

The other three languages, with more complicated morphology, yield a larger number of rules. In Italian, we rely on the sets of rules spanning: (1) regular formation of plural (libro / libri); (2) regular verb conjugation (aspettare / aspettiamo); (3) regular formation of past participle (aspettare / aspettato); and (4) rules regarding grammatical gender (bianco / bianca). Besides these, another set of rules is used for German and Russian: (5) regular declension (e.g., asiatisch / asiatischem).

**Extracting REPEL Pairs** As another source of implicit semantic signals, \( W \) also contains words which represent derivational antonyms: e.g., two words that denote concepts with opposite meanings, generated through a derivational process. We use a standard set of EN “antonymy” prefixes: \( AP_{en} = \{\text{dis, il, un, in, im, ir, mis, non, anti}\} \) (Fromkin et al., 2013). If \( w_1, w_2 \in W_{en} \), where \( w_2 \) is generated by adding a prefix from \( AP_{en} \) to \( w_1 \), then \((w_1, w_2)\) and \((w_2, w_1)\) are added to the set of REPEL constraints \( R \). This rule generates pairs such as (advantage, disadvantage) and (regular, irregular). An additional rule replaces the suffix -ful with -less, extracting antonyms such as (careful, careless).

Following the same principle, we use \( AP_{de} = \{\text{un, nicht, anti, ir, in, miss}\} \), \( AP_{it} = \{\text{in, ir, im, antil}\} \), and \( AP_{ru} = \{\text{we, amti}\} \). For instance, this generates an IT pair (rispettoso, irrispettoso) (see Fig. 1). For DE, we use another rule targeting suffix replacement: -voll is replaced by -los.

We further expand the set of REPEL constraints by transitive combining antonymy pairs from the previous step with inflectional ATTRACT pairs. This step yields additional constraints such as (rispettoso, irrispettosi) (see Fig. 1). The final \( A \) and \( R \) constraint counts are given in Tab. 3. The full sets of rules are available as supplemental material.

### 3 Experimental Setup

**Training Data and Setup** For each of the four languages we train the skip-gram with negative sampling (SGNS) model (Mikolov et al., 2013)
on the latest Wikipedia dump of each language. We induce 300-dimensional word vectors, with the frequency cut-off set to 10. The vocabulary sizes \(||W|||\) for each language are provided in Tab. 3.\(^6\) We label these collections of vectors SGNS-LARGE.

### Other Starting Distributional Vectors

We also analyse the impact of *morph-fitting* on other collections of well-known EN word vectors. These vectors have varying vocabulary coverage and are trained with different architectures. We test standard distributional models; Common-Crawl GloVe (Pennington et al., 2014), SGNS vectors (Mikolov et al., 2013) with various contexts (BOW = bag-of-words; DEPS = dependency contexts), and training data (PW = Polyglot Wikipedia from Al-Rfou et al. (2013); 8B = 8 billion token word2vec corpus), following (Levy and Goldberg, 2014) and (Schwartz et al., 2015). We also test the symmetric-pattern based vectors of Schwartz et al. (2016) (SymPat-Emb), count-based PMI-weighted vectors reduced by SVD (Baroni et al., 2014) (Count-SVD), a model which replaces the context modelling function from CBOW with bidirectional LSTMs (Melamud et al., 2016) (Context2Vec), and two sets of EN vectors trained by injecting multilingual information: BiSkip (Luong et al., 2015) and MultiCCA (Faruqui and Dyer, 2014).

We also experiment with standard well-known distributional spaces in other languages (IT and DE), available from prior work (Diu et al., 2015; Luong et al., 2015; Vulić and Korhonen, 2016a).

### Morph-fixed Vectors

A baseline which utilises an equal amount of knowledge as morph-fitting, termed *morph-fixing*, fixes the vector of each word to the distributional vector of its most frequent inflectional synonym, tying the vectors of low-frequency words to their more frequent inflections. For each word \(w_1\), we construct a set of \(M + 1\) words \(W_{w_1} = \{w_1, w'_1, \ldots, w'_M\}\) consisting of the word \(w_1\) itself and all \(M\) words which co-occur with \(w_1\) in the ATTRACT constraints. We then choose the word \(w'_{\max}\) from the set \(W_{w_1}\) with the maximum frequency in the training data, and fix all other word vectors in \(W_{w_1}\) to its word vector. The morph-fixed vectors (MFtX) serve as our primary baseline, as they outperformed another straightforward baseline based on stemming across all of our intrinsic and extrinsic experiments.

### Morph-fitting Variants

We analyse two variants of morph-fitting: (1) using ATTRACT constraints only (MFtX-A), and (2) using both ATTRACT and REPEL constraints (MFtX-AR).

### 4 Intrinsic Evaluation: Word Similarity

#### Evaluation Setup and Datasets

The first set of experiments intrinsically evaluates *morph-fitted* vector spaces on word similarity benchmarks, using Spearman’s rank correlation as the evaluation metric. First, we use the SimLex-999 dataset, as well as SimVerb-3500, a recent EN verb pair similarity dataset providing similarity ratings for 3,500 verb pairs.\(^7\) SimLex-999 was translated to DE, IT, and RU by Leviant and Reichart (2015), and they crowdsourced similarity scores from native speakers. We use this dataset for our multilingual evaluation.\(^8\)

### Morph-fitting EN Word Vectors

As the first experiment, we morph-fit a wide spectrum of EN distributional vectors induced by various architectures (see Sect. 3). The results on SimLex and SimVerb are summarised in Tab. 4. The results with EN SGNS-LARGE vectors are shown in Fig. 3a. Morph-fitted vectors bring consistent improvement across all experiments, regardless of the quality of the initial distributional space. This finding confirms that the method is robust: its effectiveness does not depend on the architecture used to construct the initial space. To illustrate the improvements, note that the best score on SimVerb for a model trained on running text is achieved by Context2vec (\(\rho = 0.388\)); injecting morphological constraints into this vector space results in a gain of 7.1 \(\rho\) points.

### Experiments on Other Languages

We next extend our experiments to other languages, testing both morph-fitting variants. The results are summarised in Tab. 5, while Fig. 3a-3d show results for the morph-fitted SGNS-LARGE vectors. These scores confirm the effectiveness and robustness of morph-fitting across languages, suggesting that the idea of fitting to morphological constraints is indeed language-agnostic, given the set of language-specific rule-based constraints. Fig. 3 also demonstrates the effectiveness of the method across different architectures and languages.

---

\(^6\) Other SGNS parameters were set to standard values (Baroni et al., 2014; Vulić and Korhonen, 2016b): 15 epochs, 15 negative samples, global learning rate: .025, subsampling rate: \(1e - 4\). Similar trends in results persist with \(d = 100, 500\).

\(^7\) Unlike other gold standard resources such as WordSim-353 (Finkelstein et al., 2002) or MEN (Bruni et al., 2014), SimLex and SimVerb provided explicit guidelines to discern between semantic similarity and association, so that related but non-similar words (e.g. *cup* and *coffee*) have a lower rating.

\(^8\) Since Leviant and Reichart (2015) re-scored the original EN SimLex, we use their EN SimLex version for consistency.
Table 4: The impact of morph-fitting (MFit-AR used) on a representative set of EN vector space models. All results show the Spearman’s ρ correlation before and after morph-fitting. The numbers in parentheses refer to the vector dimensionality.

<table>
<thead>
<tr>
<th>Vectors</th>
<th>Evaluation</th>
<th>SimLex-999</th>
<th>SimVerb-3500</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. SG-BOW2-PW (300)</td>
<td>Distrib.</td>
<td>.324</td>
<td>.376</td>
</tr>
<tr>
<td>(Mikolov et al., 2013)</td>
<td>MFit-A</td>
<td>.339</td>
<td>.423</td>
</tr>
<tr>
<td>2. GloVe-6B (300)</td>
<td></td>
<td>.324</td>
<td>.376</td>
</tr>
<tr>
<td>(Pennington et al., 2014)</td>
<td>MFit-AR</td>
<td>.339</td>
<td>.423</td>
</tr>
<tr>
<td>3. Count-SVD (500)</td>
<td></td>
<td>.267</td>
<td>.360</td>
</tr>
<tr>
<td>(Baroni et al., 2014)</td>
<td></td>
<td>.199</td>
<td>.301</td>
</tr>
<tr>
<td>4. SG-DEPS-PW (300)</td>
<td></td>
<td>.376</td>
<td>.434</td>
</tr>
<tr>
<td>(Levy and Goldberg, 2014)</td>
<td></td>
<td>.313</td>
<td>.418</td>
</tr>
<tr>
<td>5. SG-DEPS-8B (500)</td>
<td></td>
<td>.373</td>
<td>.441</td>
</tr>
<tr>
<td>(Lan et al., 2014)</td>
<td></td>
<td>.356</td>
<td>.473</td>
</tr>
<tr>
<td>6. MultiCCA-EN (512)</td>
<td></td>
<td>.314</td>
<td>.391</td>
</tr>
<tr>
<td>(Faruqui and Dyer, 2014)</td>
<td></td>
<td>.296</td>
<td>.354</td>
</tr>
<tr>
<td>7. BiSkip-EN (256)</td>
<td></td>
<td>.276</td>
<td>.356</td>
</tr>
<tr>
<td>(Luong et al., 2015)</td>
<td></td>
<td>.260</td>
<td>.333</td>
</tr>
<tr>
<td>8. SG-BOW2-8B (500)</td>
<td></td>
<td>.373</td>
<td>.440</td>
</tr>
<tr>
<td>(Schwartz et al., 2015)</td>
<td></td>
<td>.348</td>
<td>.441</td>
</tr>
<tr>
<td>9. SymPat-Emb (500)</td>
<td></td>
<td>.381</td>
<td>.442</td>
</tr>
<tr>
<td>(Schwartz et al., 2016)</td>
<td></td>
<td>.284</td>
<td>.373</td>
</tr>
<tr>
<td>10. Context2Vec (600)</td>
<td></td>
<td>.371</td>
<td>.440</td>
</tr>
<tr>
<td>(Melandri et al., 2016)</td>
<td></td>
<td>.388</td>
<td>.459</td>
</tr>
</tbody>
</table>

Table 5: Results on multilingual SimLex-999 (EN, DE, and IT) with two morph-fitting variants.

<table>
<thead>
<tr>
<th>Vectors</th>
<th>Evaluation</th>
<th>SimLex-999</th>
<th>SimVerb-3500</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN: GloVe-6B (300)</td>
<td>Distrib.</td>
<td>.324</td>
<td>.376</td>
</tr>
<tr>
<td>(Mikolov et al., 2013)</td>
<td>MFit-A</td>
<td>.339</td>
<td>.423</td>
</tr>
<tr>
<td>EN: SG-BOW2-PW (300)</td>
<td></td>
<td>.267</td>
<td>.360</td>
</tr>
<tr>
<td>(Mikolov et al., 2013)</td>
<td>MFit-AR</td>
<td>.354</td>
<td>.414</td>
</tr>
<tr>
<td>DE: SG-DEPS-PW (300)</td>
<td></td>
<td>.267</td>
<td>.318</td>
</tr>
<tr>
<td>(Vulić and Korhonen, 2016a)</td>
<td></td>
<td>.325</td>
<td>.421</td>
</tr>
<tr>
<td>DE: BiSkip-DE (256)</td>
<td></td>
<td>.354</td>
<td>.414</td>
</tr>
<tr>
<td>(Luong et al., 2015)</td>
<td></td>
<td>.354</td>
<td>.414</td>
</tr>
<tr>
<td>IT: SG-DEPS-PW (300)</td>
<td></td>
<td>.237</td>
<td>.351</td>
</tr>
<tr>
<td>(Vulić and Korhonen, 2016a)</td>
<td></td>
<td>.391</td>
<td>.446</td>
</tr>
<tr>
<td>IT: CBOW5-Wacky (300)</td>
<td></td>
<td>.363</td>
<td>.417</td>
</tr>
<tr>
<td>(Dinu et al., 2015)</td>
<td></td>
<td>.363</td>
<td>.417</td>
</tr>
</tbody>
</table>

Figure 2: A comparison of morph-fitting (the MFit-AR variant) with two other standard specialisation approaches using the same set of morphological constraints: Retrofitting (RF) (Faruqui et al., 2015) and Counter-fitting (CF) (Mrkšić et al., 2016). Spearman’s ρ correlation scores on the multilingual SimLex-999 dataset for the same six distributional spaces from Tab. 5.

Comparison to Other Specialisation Methods

We also tried using other post-processing specialisation models from the literature in lieu of ATTRACT-REPEL using the same set of “morphological” synonymy and antonymy constraints. We compare ATTRACT-REPEL to the retrofitting model of (Faruqui et al., 2015) and counter-fitting (Mrkšić et al., 2017a). The two baselines were trained for 20 iterations using suggested settings. The results for EN, DE, and IT are summarised in Fig. 2. They clearly indicate that MFit-AR outperforms the two other post-processors for each language. We hypothesise that the difference in performance mainly stems from context-sensitive vector space updates performed by ATTRACT-REPEL. Conversely, the other two models perform pairwise updates which do not consider what effect each update has on the example pair’s relation to other word vectors (for a detailed comparison, see (Mrkšić et al., 2017b)).

Besides their lower performance, the two other specialisation models have additional disadvantages compared to the proposed morph-fitting model. First, retrofitting is able to incorporate only synonymy/ATTRACT pairs, while our results demonstrate the usefulness of both types of constraints, both for intrinsic evaluation (Tab. 5) and downstream tasks (see later Fig. 3). Second, counter-fitting is computationally intractable with SGNS-LARGE vectors, as its regularisation term involves the computation of all pairwise distances between words in the vocabulary.

Further Discussion

The simplicity of the used language-specific rules does come at a cost of occasionally generating incorrect linguistic constraints such as (tent, intend), (prove, improve) or (press, impress). In future work, we will study how to fur-
ther refine extracted sets of constraints. We also plan to conduct experiments with gold standard morphological lexicons on languages for which such resources exist (Sylak-Glassman et al., 2015; Cotterell et al., 2016b), and investigate approaches which learn morphological inflections and derivations in different languages automatically as another potential source of morphological constraints (Soricut and Och, 2015; Cotterell et al., 2016a; Faruqui et al., 2016; Kann et al., 2017; Aharoni and Goldberg, 2017, i.a.).

5 Downstream Task: Dialogue State Tracking (DST)

Goal-oriented dialogue systems provide conversational interfaces for tasks such as booking flights or finding restaurants. In slot-based systems, application domains are specified using ontologies that define the search constraints which users can express. An ontology consists of a number of slots and their assorted slot values. In a restaurant search domain, sets of slot-values could include \text{PRICE} = [\text{cheap}, \text{expensive}] or \text{FOOD} = [\text{Thai, Indian, ...}].

The DST model is the first component of modern dialogue pipelines (Young, 2010). It serves to capture the intents expressed by the user at each dialogue turn and update the belief state. This probability distribution over the possible dialogue states (defined by the domain ontology) is the system’s internal estimate of the user’s goals. It is used by the downstream dialogue manager component to choose the subsequent system response (Su et al., 2016). The following example shows the true dialogue state in a multi-turn dialogue:

\begin{verbatim}
User: What’s good in the southern part of town?
    inform(area=south)
System: Vedanta is the top-rated Indian place.
    User: How about something cheaper?
    inform(area=south, price=cheap)
System: Seven Days is very popular. Great hot pot.
    User: What’s the address?
    inform(area=south, price=cheap); request(address)
System: Seven Days is at 66 Regent Street.
\end{verbatim}

The Dialogue State Tracking Challenge (DSTC) shared task series formalised the evaluation and provided labelled DST datasets (Henderson et al., 2014a,b; Williams et al., 2016). While a plethora of DST models are available based on, e.g., handcrafted rules (Wang et al., 2014) or conditional random fields (Lee and Eskkenazi, 2013), the recent DST methodology has seen a shift towards neural-network architectures (Henderson et al., 2014c,d; Zilka and Jurcicek, 2015; Mrkšić et al., 2015; Perez and Liu, 2017; Liu and Perez, 2017; Vodolán et al., 2017; Mrkšić et al., 2017a, i.a.).

Model: Neural Belief Tracker To detect intents in user utterances, most existing models rely on either (or both): 1) Spoken Language Understanding models which require large amounts of annotated training data; or 2) hand-crafted, domain-specific lexicons which try to capture lexical and morphological variation. The Neural Belief Tracker (NBT) is a novel DST model which overcomes both issues by reasoning purely over pre-trained word vectors (Mrkšić et al., 2017a). The NBT learns to compose these vectors into intermediate utterance and context representations. These are then used to decide which of the ontology-defined intents (goals) have been expressed by the user. The NBT model keeps word vectors fixed during training, so that unseen, yet related words can be mapped to the right intent at test time (e.g. northern to north).

Data: Multilingual WOZ 2.0 Dataset Our DST evaluation is based on the WOZ dataset, released by Wen et al. (2017). In this Wizard-of-Oz setup, two Amazon Mechanical Turk workers assumed the role of the user and the system asking/providing information about restaurants in Cambridge (operating over the same ontology and database used for DSTC2 (Henderson et al., 2014a)). Users typed instead of speaking, removing the need to deal with noisy speech recognition. In DSTC datasets, users would quickly adapt to the system’s inability to deal with complex queries. Conversely, the WOZ setup allowed them to use sophisticated language. The WOZ 2.0 release expanded the dataset to 1,200 dialogues (Mrkšić et al., 2017a). In this work, we use translations of this dataset to Italian and German, released by Mrkšić et al. (2017b).

Evaluation Setup The principal metric we use to measure DST performance is the joint goal accuracy, which represents the proportion of test set dialogue turns where all user goals expressed up to that point of the dialogue were decoded correctly (Henderson et al., 2014a). The NBT models for EN, DE and IT are trained using four variants of the SGNS-LARGE vectors: 1) the initial distributional vectors; 2) morph-fixed vectors; 3) and 4) the two variants of morph-fitted vectors (see Sect. 3).

As shown by Mrkšić et al. (2017b), semantic specialisation of the employed word vectors ben-
efits DST performance across all three languages. However, large gains on SimLex-999 do not always induce correspondingly large gains in downstream performance. In our experiments, we investigate the extent to which morph-fitting improves DST performance, and whether these gains exhibit stronger correlation with intrinsic performance.

**Results and Discussion** The dark bars (against the right axes) in Fig. 3 show the DST performance of NBT models making use of the four vector collections. IT and DE benefit from both kinds of morph-fitting: IT performance increases from 74.1 → 78.1 (MFit-A) and DE performance rises even more: 60.6 → 66.3 (MFit-AR), setting a new state-of-the-art score for both datasets. The morph-fixed vectors do not enhance DST performance, probably because fixing word vectors to their highest frequency inflectional form eliminates useful semantic content encoded in the original vectors. On the other hand, morph-fitting makes use of this information, supplementing it with semantic relations between different morphological forms. These conclusions are in line with the SimLex gains, where morph-fitting outperforms both distributional and morph-fixed vectors.

English performance shows little variation across the four word vector collections investigated here. This corroborates our intuition that, as a morphologically simpler language, English stands to gain less from fine-tuning the morphological variation for downstream applications. This result again points at the discrepancy between intrinsic and extrinsic evaluation: the considerable gains in SimLex performance do not necessarily induce similar gains in downstream performance. Additional discrepancies between SimLex and downstream DST performance are detected for German and Italian. While we observe a slight drop in SimLex performance with the DE MFit-AR vectors compared to the MFit-A ones, their relative performance is reversed in the DST task. On the other hand, we see the opposite trend in Italian, where the MFit-A vectors score lower than the MFit-AR vectors on SimLex, but higher on the DST task. In summary, we believe these results show that SimLex is not a perfect proxy for downstream performance in language understanding tasks. Regardless, its performance does correlate with downstream performance to a large extent, providing a useful indicator for the usefulness of specific word vector
spaces for extrinsic tasks such as DST.

6 Related Work

Semantic Specialisation A standard approach to incorporating external information into vector spaces is to pull the representations of similar words closer together. Some models integrate such constraints into the training procedure, modifying the prior or the regularisation (Yu and Dredze, 2014; Xu et al., 2014; Bian et al., 2014; Kiela et al., 2015), or using a variant of the SGNS-style objective (Liu et al., 2015; Osborne et al., 2016). Another class of models, popularly termed retrofitting, injects lexical knowledge from available semantic databases (e.g., WordNet, PPDB) into pre-trained word vectors (Faruqui et al., 2015; Jauhar et al., 2015; Wieting et al., 2015; Nguyen et al., 2016; Mrkšić et al., 2016). Morph-fitting falls into the latter category. However, instead of resorting to curated knowledge bases, and experimenting solely with English, we show that the morphological richness of any language can be exploited as a source of inexpensive supervision for fine-tuning vector spaces, at the same time specialising them to better reflect true semantic similarity, and learning more accurate representations for low-frequency words.

Word Vectors and Morphology The use of morphological resources to improve the representations of morphemes and words is an active area of research. The majority of proposed architectures encode morphological information, provided either as gold standard morphological resources (Sylak-Glassman et al., 2015) such as CELEX (Baayen et al., 1995) or as an external analyser such as Morfessor (Creutz and Lagus, 2007), along with distributional information jointly at training time in the language modelling (LM) objective (Luong et al., 2013; Botha and Blunsom, 2014; Qiu et al., 2014; Cotterell and Schütze, 2015; Bhatia et al., 2016, i.a.). The key idea is to learn a morphological composition function (Lazaridou et al., 2013; Cotterell and Schütze, 2017) which syntheisises the representation of a word given the representations of its constituent morphemes. Contrary to our work, these models typically coalesce all lexical relations.

Another class of models, operating at the character level, shares a similar methodology: such models compose token-level representations from subcomponent embeddings (subwords, morphemes, or characters) (dos Santos and Zadrozny, 2014; Ling et al., 2015; Cao and Rei, 2016; Kim et al., 2016; Wieting et al., 2016; Verwimp et al., 2017, i.a.).

In contrast to prior work, our model decouples the use of morphological information, now provided in the form of inflectional and derivational rules transformed into constraints, from the actual training. This pipelined approach results in a simpler, more portable model. In spirit, our work is similar to Cotterell et al. (2016b), who formulate the idea of post-training specialisation in a generative Bayesian framework. Their work uses gold morphological lexicons; we show that competitive performance can be achieved using a non-exhaustive set of simple rules. Our framework facilitates the inclusion of antonyms at no extra cost and naturally extends to constraints from other sources (e.g., WordNet) in future work. Another practical difference is that we focus on similarity and evaluate morph-fitting in a well-defined downstream task where the artefacts of the distributional hypothesis are known to prompt statistical system failures.

7 Conclusion and Future Work

We have presented a novel morph-fitting method which injects morphological knowledge in the form of linguistic constraints into word vector spaces. The method makes use of implicit semantic signals encoded in inflectional and derivational rules which describe the morphological processes in a language. The results in intrinsic word similarity tasks show that morph-fitting improves vector spaces induced by distributional models across four languages. Finally, we have shown that the use of morph-fitted vectors boosts the performance of downstream language understanding models which rely on word representations as features, especially for morphologically rich languages such as German.

Future work will focus on other potential sources of morphological knowledge, porting the framework to other morphologically rich languages and downstream tasks, and on further refinements of the post-processing specialisation algorithm and the constraint selection.

Acknowledgments

This work is supported by the ERC Consolidator Grant LEXICAL: Lexical Acquisition Across Languages (no 648909). RR is supported by the Intel-ICRI grant: Hybrid Models for Minimally Supervised Information Extraction from Conversations. The authors are grateful to the anonymous reviewers for their helpful suggestions.
References


Martin Haspelmath and Andrea Sims. 2013. Understanding morphology.


Morphological Rules

In this supplemental material, we provide a short comprehensive overview of simple language-specific morphological rules in English (EN), German (DE), Italian (IT), and Russian (RU). These rules were used to build the sets of synonymous ATTRACT and antonymous REPEL constraints for our morph-fitting fine-tuning procedure. As discussed in the paper, the linguistic constraints extracted from the rules require only a comprehensive list of vocabulary words in each language. A native speaker of each language used in our experiments is able to easily come up with these sets of morphological rules (or at least with a reasonable subset of rules) without any linguistic training. What is more, the rules for German, Italian, and Russian were created by non-native and non-fluent speakers who have only a passive or limited knowledge of each language from scratch. These rules were additionally used to generate the Morph-SimLex evaluation set, also provided as supplemental material.

Different languages differ in their “morphological richness” (e.g., declension, verb conjugation, plural forming, gender) which consequently leads to the varying number of rules in each language. However, all four languages in our study display morphological regularities described by simple morphological rules that are exploited to build sets of ATTRACT and REPEL linguistic constraints in each language from scratch.\(^9\)

Vocabularies \(W\) in all four languages are labeled \(W_{en}, W_{de}, W_{it}, W_{ru}\). We add the pairs \((w_1, w_2)\) and \((w_2, w_1)\) generated by the rules to the sets of constraints iff both \(w_1, w_2 \in W\). After we generate all such constraints, since some constraints may have been generated by more than one rule, we remove all duplicates from the respective sets of ATTRACT and REPEL constraints.

Before we start, we will define two simple functions: (i) the function \(w([::1-N])\) strips the last \(N\) characters from the word \(w\), (ii) the function \(w.ew('sub')\) tests if the word \(w\) ends with a sequence of characters \(sub\). For instance, \(create[::1]\) returns \(creat\) while \(create.ew('s')\) returns \(False\) and \(create.ew('e')\) returns \(True\).

English Rules

Inflectional Synonymy: ATTRACT As discussed in the paper, we rely on only two simple inflectional morphological rules in English: - \(w_2 = w_1 + 's'/'ed'/'ing'\). This rule yields constraints such as (\textit{sing}, speaking), (turtle, turtles), or (clean, cleaned).
- If \(w_1.ew('e')\), then \(w_2 = w_1[::1] + 'ed'/'ing'\). This rule yields constraints such as (create, creating), or (\textit{generate}, generated).

Derivational Antonymy: REPEL We assume the following set of standard “antonymy” prefixes in English: \(AP_{en} = \{ 'dis$', 'il', 'un', 'in', 'im', 'ir', 'mis', 'non', 'anti' \}. We rely on the following derivational rules to extract REPEL pairs:
- \(w_2 = ap + w_1\), where \(ap \in AP_{en}\). This rule yields constraints such as (\textit{improve}, immature), (allow, disallow) or (regularity, irregularity).
- If \(w_1.ew('al')\), then \(w_2 = w_1[::1] + 's'\). This rule yields constraints such as (cheerful, cheerless).

As mentioned in the paper, for all four languages we further expand the set of REPEL constraints by transitively combining antonymy pairs with inflectional ATTRACT pairs. In simple words, the friend of my enemy is my enemy. This means that, given an ATTRACT pair (allow, allows) and a REPEL pair (allow, disallow), we extract another REPEL pair (allows, disallow).

German Rules

Inflectional Synonymy: ATTRACT Being morphologically richer than English, the German language naturally requires more rules to describe its (inflectional) morphological richness and variation. First, we capture the regular declension of nouns and adjectives by the following heuristic:
- Generate a set of words \(W_{w_1} = \{w_1, w_2 |w_2 = w_1 + 'e'/'em'/'en'/'er'/'es'\}\): take the Cartesian product on \(W_{w_1} \times W_{w_1}\) and then exclude \((w_1, w_1)\)

\(^9\)Note that the rules for extracting ATTRACT constraints were additionally used to generate the Morph-SimLex evaluation set, also provided as supplemental material.
pairs with identical words. This rule generates pairs such as (schottisch, schottische), (schottischem, schottischen).

The second set of rules describes regular verb morphology, i.e., verb conjugation in the present and past tense, and the formation of regular past participles. This set of rules may be expressed as:
- If \( w_1.\text{ew}('\text{en}') \), then \( w'_1 = w_1[:: -2] \).
- If \( w'_1.\text{ew}('\text{t}') \), then generate a set of words \( W_{w_1} = \{ w_1, w_2 | w_2 = w_1 + '\text{et}'/\text{et} & \}$, \( w_2 = '\text{ge'} + w'_1 + '\text{et}' \}$, else (if not \( w'_1.\text{ew}('\text{t}') \)), generate a set of words \( W_{w_1} = \{ w_1, w_2 | w_2 = w_1 + '\text{et}'/\text{et} & \}$, \( w_2 = '\text{ge'} + w'_1 + '\text{et}' \}$). We then take the Cartesian product on \( W_{w_1} \times W_{w_1} \).

Again, all pairs with identical words were discarded. This rule yields pairs such as (machen, machten), (kaufst, kauft), or (arbeitete, arbeitete) and (arbeiteten, arbeiteten).

Another set of rules targets the regular formation of plural nouns:
- If \( w_1.\text{ew}('\text{ei}') \) or \( w_1.\text{ew}('\text{eit}') \) or \( w_1.\text{ew}('\text{eicht}') \) or \( w_1.\text{ew}('\text{eicht}') \) or \( w_1.\text{ew}('\text{ung}') \), then \( w_2 = w_1 + '\text{en}' \). This rule results in pairs such as (wahrheit, wahrheiten) or (gemeinschaft, gemeinschaften).
- If \( w_1.\text{ew}('\text{in}') \), then \( w_2 = w_1 + '\text{en}' \). This rule generates pairs such as (lektorin, lektorinnen) or (lehrerin, lehrerinnen).
- If \( w_1.\text{ew}('\text{er}') \) or \( w_1.\text{ew}('\text{ei}') \) or \( w_1.\text{ew}('\text{e}') \) then \( w_2 = w_1 + '\text{e}' \). This rule results pairs such as (auto, autos).

Derivation Antonymy: REPEL We assume the following set of standard “antonymy” prefixes in German: \( AP_{de} = \{ '\text{un}', '\text{nicht}', '\text{anti}', '\text{ir}', '\text{in}', '\text{miss}' \} \). We rely on the following derivational rules to extract REPEL pairs in German:
- \( w_2 = \text{ap} + w_1 \), where \( \text{ap} \in AP_{de} \). This rule yields constraints such as (aktiv, inaktiv), (wandelbar, unwandelbar) or (zyklone, antizyklide).

The set of REPEL is then again transitively expanded yielding pairs such as (relevant, irrelevanter) or (aktivem, inaktiv).

Italian Rules

Inflectional Synonymy: ATTRACT The first set of rules aims at capturing the regular plural forming in Italian (e.g., libro, libri) and regular differences in gender (e.g., rapido, rapida). We rely on the simple heuristic which can be expressed as follows:
- If \( w_1.\text{ew}('\text{a}'/\text{e}'/\text{o}'/\text{i}') \), then generate a set of words \( W_{w_1} = \{ w_2 | w_2 = w_1[:: -1] + 'a'/\text{e}'/\text{o}'/\text{i}' \}$, and take the Cartesian product on \( W_{w_1} \times W_{w_1} \) discard pairs with identical words. This rule yields pairs such as (nero, neri) or (generazione, generazioni).
- If \( w_1.\text{ew}('\text{ca}') \), then \( w_2 = w_1 + '\text{e}' \). This rule generates pairs such as (tartaruga, tartarughe) or (bianca, bianche).

The second set of rules targets regular verb conjugation in Italian and the formation of regular past participles. The following rules are used:
- If \( w_1.\text{ew}('\text{are}') \), then generate a set of words \( W_{w_1} = \{ w_2 | w_2 = w_1[:: -1] + '\text{ame}'/\text{ato} \}$, then \( W_{w_1} = \{ w_2 | w_2 = w_1[:: -1] + '\text{ame}'/\text{ato} \}$, then \( W_{w_1} = \{ w_2 | w_2 = w_1[:: -1] + '\text{ame}'/\text{ato} \}$). We take the Cartesian product on \( W_{w_1} \times W_{w_1} \) discard pairs with identical words. This rule results in pairs such as (aspettare, aspettiamo).
- If \( w_1.\text{ew}('\text{ere}') \), then generate a set of words \( W_{w_1} = \{ w_2 | w_2 = w_1[:: -1] + '\text{amo}'/\text{ato} \}$, then \( W_{w_1} = \{ w_2 | w_2 = w_1[:: -1] + '\text{amo}'/\text{ato} \}$, then \( W_{w_1} = \{ w_2 | w_2 = w_1[:: -1] + '\text{amo}'/\text{ato} \}$). We take the Cartesian product on \( W_{w_1} \times W_{w_1} \) discard pairs with identical words. This rule results in pairs such as (ricevere, ricevete) or (riceve, ricevuto).

Derivation Antonymy: REPEL We assume the following set of standard “antonymy” prefixes in Italian: \( AP_{it} = \{ '\text{in}', '\text{ir}', '\text{im}', '\text{anti}' \} \). The following derivational rule is used to extract REPEL pairs:
- \( w_2 = \text{ap} + w_1 \), where \( \text{ap} \in AP_{it} \). This rule yields constraints such as (attivo, inattivo) or (rispettosa, irrispettosa).
The set of REPEL was then expanded as before, e.g., with additional pairs such as (rispettosa, irrispettosi) generated.

Russian Rules

Inflectional Synonymy: ATTRACT The first set of rules in Russian targets the regular forming of plural in Russian. A few simple heuristics are used as follows:
- \( w_2 = w_1 + 'и' / 'ы' \). This rule yields pairs such as (альбом, альбомы), transliterated as: (al’bom, al’bomy).
- if \( w_1.ew('а' / 'я' / 'ь') \), then \( w_2 = w_1[-1] + 'и' / 'ы' \). This rule generates pairs such as (письма, письма) or (pis’ma, pis’ma).
- if \( w_1.ew('о') \), then \( w_2 = w_1[-1] + 'а' \). This rule generates pairs such as (письма, письма).
- if \( w_1.ew('е') \), then \( w_2 = w_1[-1] + 'я' \). This rule generates pairs such as (пластырь, пластыря).

The next set of rules targets regular verb conjugation of Russian verbs as well as the regular formation of past participles. We again build a simple heuristic to extract ATTRACT pairs:
- if \( w_1.ew('т'и' / 'ть') \), then generate a set of words \( W_{w_1} = \{w_1, w_2 | w_2 = w_1[-2] + 'ы' / 'о' / 'и' / 'и' / 'н' / 'е' / 'е' / 'м' / 'й', w_2 = w_1[-2] + 'е' / 'е' / 'р' / 'р' / 'ы' / 'ы' / 'я', w_2 = w_1[-2] + 'и' / 'я' / 'а' / 'а' \} \) and take the Cartesian product on \( W_{w_1} \times W_{w_1} \) discarding pairs with identical words. This rule yields pairs such as (работам, работами): (работам, работами).

Yet another set of rules targets regular adjective comparison and gender:
- if \( w_1.ew('ый') / 'оий' / 'ий')', then generate a set of words \( W_{w_1} = \{w_1, w_2 | w_2 = w_1[-2] + 'е' / 'и' / 'и' / 'й' \} \). This rule yields pairs such as (быстрый, быстро): (bystrij, bystree).
- if \( w_1.ew('ая') \), then generate a set of words \( W_{w_1} = \{w_1, w_2 | w_2 = w_1[-2] + 'е' / 'и' / 'и' / 'й' \} \). This rule yields pairs such as (новая, новые): (novaja, novye).
- if \( w_1.ew('ое') \), then generate a set of words \( W_{w_1} = \{w_1, w_2 | w_2 = w_1[-2] + 'и' / 'и' / 'й' \} \). This rule yields pairs such as (новое, новый): (novoe, novyj).

Derivational Antonymy: REPEL We assume the following set of standard “antonymy” prefixes in Russian: \( AP_{ru} = \{не, анти\} \), and simply use the following rule:
- \( w_2 = ap + w_1 \), where \( ap \in AP_{ru} \). This rule yields constraints such as (адекватный, неадекватный) or (вирусная, антивирусная), translected as: (adekvatnyj, neadekvatnyj) and (virusnaja, antivirusnaja).

The further expansion of REPEL constraints yields pairs such as (адекватный, неадекватный): (adekvatnyj, neadekvatnaja).

Further Discussion

We stress that the listed rules for all four languages are non-exhaustive and do not cover all possible inflectional and derivational morphological phenomena. More linguistic constraints may be extracted by resorting to more sophisticated rules covering finer-grained morphological processes (e.g., covering irregular plural forming or irregular verb conjugation and past participle forming, or non-standard declensions). Further, the listed rules, written by non-native speakers without any linguistic training in a very short time span, do not necessarily rely on established linguistic theories in each language, but are rather simple heuristics aiming to capture morphological regularities.