Wiktionary-Based Word Embeddings

Gerard de Melo gdm@demelo.org
IIS, Tsinghua University, Beijing, China

Abstract

Vectorial representations of words have grown remarkably popular in natural language processing and machine translation. The recent surge in deep learning-inspired methods for producing distributed representations has been widely noted even outside these fields. Existing representations are typically trained on large monolingual corpora using context-based prediction models. In this paper, we propose extending pre-existing word representations by exploiting Wiktionary. This process results in a substantial extension of the original word vector representations, yielding a large multilingual dictionary of word embeddings. We believe that this resource can enable numerous monolingual and cross-lingual applications, as evidenced in a series of monolingual and cross-lingual semantic experiments that we have conducted.

1 Introduction

Vectorial representations of words have grown to play an important role in natural language processing and machine translation. Especially for the latter, deep learning and representation learning-based approaches have recently proven remarkably effective (Sutskever et al., 2014; Luong et al., 2015). Vector-based encodings of meaning are a central ingredient in many of these recent neural machine translation systems, although they can also be beneficial in ordinary phrase-based machine translation (Mikolov et al., 2013b).

In this work, we focus on the task of creating vector representations of multilingual words (as well as lexicalized phrases). Previous work in this area has relied on multilingual corpora to train bilingual word vectors. We investigate to what extent external large-scale resources can be used to create much more multilingual word representation data. In particular, we rely on Wiktionary, a sister project of Wikipedia that for many years now has been creating a large, collaboratively edited online dictionary. Due to its rich multilingual data, now with over 4 million entries in over 1,000 languages, Wiktionary has been used extensively in natural language processing, e.g. for part-of-speech tagging (Li et al., 2012) and named entity recognition (Richman and Schone, 2008), for cross-language image search (Etzioni et al., 2007) and text classification (Nastase and Strapparava, 2013), and for producing language registries (de Melo, 2015) and etymological databases (de Melo, 2014). Wiktionary has also made it possible to translate lexical knowledge bases such as WordNet to hundreds of languages (de Melo and Weikum, 2009) and to translate thesauri (Borin et al., 2014). Finally, it has been used as an extra ingredient in regular machine translation systems (Göhring, 2014).

Relying on Wiktionary instead of on other training data has playfully been called Wiki-ly supervision (Li et al., 2012). Our work constitutes a form of Wiktionary-based supervision

This research was supported by China 973 Program Grants 2011CBA00300, 2011CBA00301, and NSFC Grants 61033001, 61361136003, 61450110088.

for multilingual word representation learning. More specifically, our method starts with existing word representations such as the widely available ones trained on large English corpora (Mikolov et al., 2013a; Pennington et al., 2014). It then uses Wiktionary to decide how to place new words into the same vector space.

2 Method

For obtaining the new word representations, we adopt the following framework. We assume, we are given vectors $\mathbf{u}_w$ for words $w \in V_0$, where V_0 is some initial vocabulary of words. Such vectors may come from any of the popular methods for training word vectors. We later use the well-known vectors from the word2vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014) projects. Our goal is to create new vectors $\mathbf{v}_w$ for all words w in a substantially larger vocabulary V, which typically will contain words from many different languages.

Note that the words w are tagged with language codes and are distinguished accordingly. For example, the Czech word *tuna* refers to a ton (the weight unit), and the Spanish word *tuna* means *prickly pear/nopal*. Neither of these bear any relationship with the fish meaning of the English word *tuna*. Thus, we treat words with different language tags as distinct entities with separate vectors. This, of course, does not preclude connections in the data from encouraging a high degree of proximity between different vectors. For example, the method will encourage the English word *sushi* to have similar vectors to those of the French and Breton *sushi*, which have the same form and meaning.

The vectors $\mathbf{v}_w$ should reflect the semantics of the words so as to be useful in downstream applications. While in the past, word vectors were often chosen such that individual dimensions have some interpretable meaning, current state-of-the-art vector space word embeddings do not have this property. Instead, we allow for words to be assigned arbitrary vectors as long as vector similarities and distances reflect corresponding word similarities and distances.

In order to achieve this, we draw on Wiktionary in order to obtain a large set W of semantic triples taking the form

$$(w_1, r, w_2)$$

where w_1, w_2 are words, and r is a relation that holds between them. The most frequent relation that we obtain from Wiktionary data is the translation relation. Other examples include synonymy and derivational relationships. Based on the triples in W, we then define the following objective:

$$\sum_{w_1} \sum_{w_2} f_W(w_1, w_2)\, \mathbf{v}_{w_1}^t \mathbf{v}_{w_2}$$

subject to

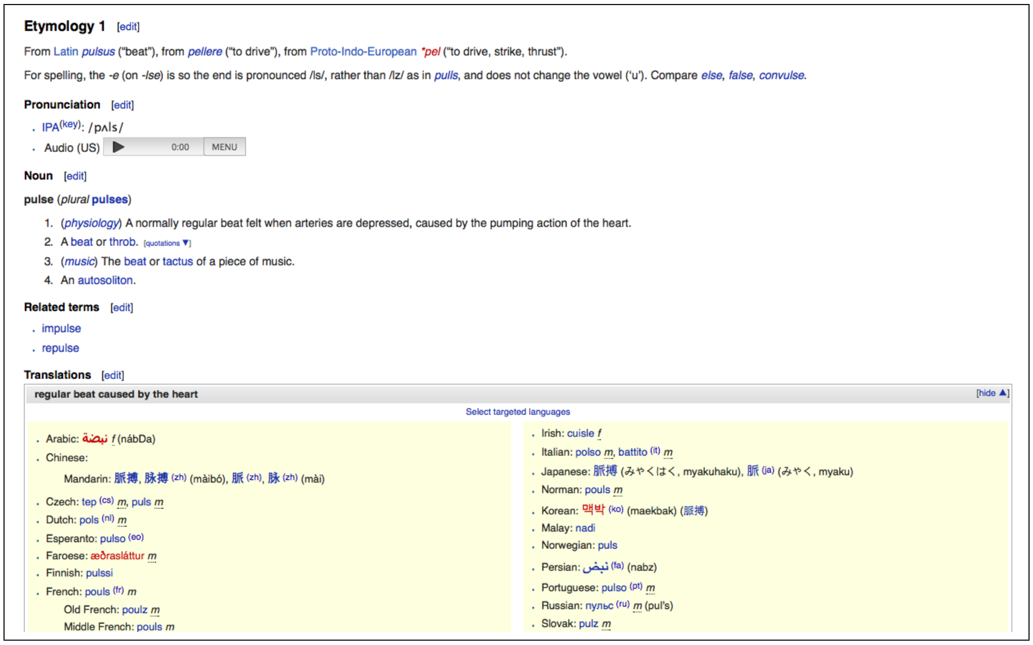
$$\|\mathbf{v}_w\|_2 \leq 1 \quad \forall w,$$

where $f_W(w_1, w_2)$ should quantify the connection strength (and polarity) between words. Thus, words are encouraged to have similarities that correspond to their relatedness, measured in terms of their dot products.

In practice, we maximize this objective function iteratively using stochastic gradient ascent. Initially, we set

$$\mathbf{v}_w = \begin{cases} \mathbf{u}_w & w \in V_0 \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{1}$$

We then repeatedly make local updates for individual triples in order to optimize the vectors in the direction of the objective. We use two different learning rates α_1, α_2 with $\alpha_1 \geq \alpha_2$ for this. The first one, α_1, is the greater of the two and is the learning rate used for new words, whilst the second, α_2, is the learning rate used for words that were already in V_0, i.e., the vocabulary



Etymology 1 [edit]

From Latin *pulsus* ("beat"), from *pellere* ("to drive"), from Proto-Indo-European **pel* ("to drive, strike, thrust").

For spelling, the *-e* (on *-lse*) is so the end is pronounced /ls/, rather than /lz/ as in *pulls*, and does not change the vowel ('u'). Compare *else*, *false*, *convulse*.

Pronunciation [edit]

- IPA$^{(key)}$: /pʌls/
- Audio (US) ▶ 0:00 MENU

Noun [edit]

pulse (*plural* **pulses**)

1. (*physiology*) A normally regular beat felt when arteries are depressed, caused by the pumping action of the heart.
2. A beat or throb. [quotations ▼]
3. (*music*) The beat or tactus of a piece of music.
4. An autosoliton.

Related terms [edit]

- impulse
- repulse

Translations [edit]

regular beat caused by the heart [hide ▲]

Select targeted languages

- Arabic: نبضة *f* (nábDa)
- Chinese:
  - Mandarin: 脈搏, 脉搏 $^{(zh)}$ (màibó), 脈 $^{(zh)}$, 脉 $^{(zh)}$ (mài)
- Czech: tep $^{(cs)}$ *m*, puls *m*
- Dutch: pols $^{(nl)}$ *m*
- Esperanto: pulso $^{(eo)}$
- Faroese: æðrasláttur *m*
- Finnish: pulssi
- French: pouls $^{(fr)}$ *m*
  - Old French: poulz *m*
  - Middle French: pouls *m*
- Irish: cuisle *f*
- Italian: polso *m*, battito $^{(it)}$ *m*
- Japanese: 脈搏 (みゃくはく, myakuhaku), 脈 $^{(ja)}$ (みゃく, myaku)
- Norman: pouls *m*
- Korean: 맥박 $^{(ko)}$ (maekbak) (脈搏)
- Malay: nadi
- Norwegian: puls
- Persian: نبض $^{(fa)}$ (nabz)
- Portuguese: pulso $^{(pt)}$ *m*
- Russian: пульс $^{(ru)}$ *m* (pul's)
- Slovak: pulz *m*

Figure 1: Wiktionary example, showing an excerpt of the page for the word *pulse*.

of the input word vectors. Since the original words have already been optimized in some prior learning process, this severely tempers the extent to which they may be negatively affected by noise, especially towards the beginning, when the vectors for the new words have not yet stabilized.

In our experiments, we simply use

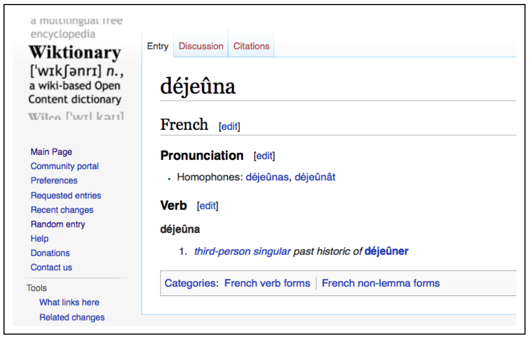
$$f_W(w_1, w_2) = |\{t \in W \mid \exists r : t = (w_1, r, w_2)\}|$$

to quantify relation strengths. While this function is non-negative, the fact that we start off with existing high-quality word vectors, that we constrain L2 norms to not grow indefinitely, and that we choose slow learning rates allow us to end up with high-quality word vectors.

3 Wiktionary Parsing

The word relationship triples in W are taken from Wiktionary. Unfortunately, Wiktionary's data is created and maintained using a rather informal semi-structured wiki markup form that is difficult to parse and not very standardized at all. For example, Figure 1 shows just a small part of the page for the word *pulse*. We rely on a custom information extraction system to produce a conversion of Wiktionary to structured data, as required for W. This is a rule-based system that partitions the raw wiki markup into different parts looking for sections and other subdivisions. It extracts translations both from the translation sections and from the gloss text, as these are a rich resource as well. In the gloss text, we sometimes have short translations, and sometimes we may also find inflectional or derivational links, as, for instance, in Figure 2.

Table 1 provides details about the extracted data, obtained by applying our parser on an XML dump of the English edition of Wiktionary (2013-12-17 version). Note that the links counts refer to the total numbers of directed links after adding inverses and removing any duplicates. We see that Wiktionary provides several million translation links as well as significant numbers of other relationships, including inflectional and derivational ones.



déjeûna

French [edit]

Pronunciation [edit]

- Homophones: déjeûnas, déjeûnât

Verb [edit]

déjeûna

1. *third-person singular past historic of* **déjeûner**

Categories: French verb forms | French non-lemma forms

Figure 2: Simpler Wiktionary example showing the page for the French word *déjeûna*, which is listed as an inflected form of *déjeûner* (to have lunch).

Item	**Count**
Translational equivalence links	3,598,807
Derivational/Inflectional links	2,455,781
Related term links	580,631
Synonymy links	490,130
Orthographic/other variant links	17,357
Unique words	3,968,843

Table 1: Wiktionary input statistics, where link counts refer to directed links after adding inverses and removing duplicates.

4 Experiments

4.1 Training

For the original input vectors, we rely on two well-known sources. The first are the pretrained word2vec vectors (Mikolov et al., 2013a) released by Google[1]. This dataset provides vector representations for words and multi-word phrases trained on a Google News dataset consisting of about 100B word tokens using word2vec. The vocabulary size is 3,000,000. However, out of these 3,000,000, actually 2,070,978 terms contain a space, most of which are multi-word expressions or named entities. Thus, the number of genuine words is much smaller.

As a second vector dataset, we experiment with the pre-trained vectors from the GloVe project (Pennington et al., 2014), which they obtained by applying their algorithm to data from a CommonCrawl corpus consisting of 840B word tokens. The vocabulary size is 2,195,960, out of which none contain a space. While the corpus is larger, it should be noted that CommonCrawl contains a lot of rather noisy Web data.

We train our model using a starting learning rate of $\alpha_1 = 0.1$ for new words and $\alpha_2 = 0.001$ for original words. The vectors stabilize fairly quickly, so we run the algorithm for only 10 epochs.

4.2 Coverage

As a result of this training process using Wiktionary data, the original word2vec representations are modified from covering 3 million tokens in just a single language to covering nearly 6

[1] `https://code.google.com/p/word2vec/`

Language	# Words (word2vec)	# Words (GloVe)
English	3,228,842	2,417,077
Italian	400,881	405,544
Latin	335,722	336,668
Spanish	242,097	242,412
French	237,189	238,744
German	113,827	114,246
Finnish	110,325	110,613
Portuguese	101,077	10,1253
Russian	67,709	67,863
Serbo-Croatian	55,652	55,778
Mandarin Chinese	50,563	50,513
Japanese	47,940	48,025
Polish	44,377	44,541
Dutch	42,808	42,900
Swedish	40,740	40,814
Hungarian	37,644	37,705
Danish	32,971	32,988
...	...	
All	5,934,987	5,133,925

Table 2: Top languages after training, using the Google word2vec and GloVe vectors as input, respectively.

million words in over 500 languages. Similarly, the GloVe vectors are extended from 2,195,960 word vectors in one language to around 5 million vectors, again in over 500 languages. In Table 2, we list the languages with the greatest coverage on the extended word2vec dataset.

Remarkably, even the coverage of English increases quite substantially, by over 200,000 entries. Although this number might appear small in comparison with the original vocabulary size of 3,000,000, there is a marked difference in quality between the two. Apart from the roughly 2 million multi-word expressions among these 3 million vocabulary items, the original data also contains vast amounts of tokens that are not genuine lexical items but simply various sorts of names, codes, misspellings, file names, and so on (e.g. *Krakowiak*, *SBSA*, *www.flu.gov*, *reccomend*, *WILLOW*). In contrast, the added vocabulary items are mostly genuine word forms, contributed by Wiktionary's editors.

Table 3 summarizes the total number of languages covered by the vectors trained on Wiktionary. A lot of rare minority languages are covered to some extent. While the vocabulary size for them tends to be small, the coverage often focuses on the most important words, such as those found in Swadesh lists and of interest in linguistic and anthropological studies. 38 languages are covered with a vocabulary size of at least 10,000. For these languages, the coverage should suffice for certain NLP tasks, including cross-lingual ones. This is what we shall investigate next.

4.3 Semantic Relatedness

Semantic relatedness studies have a long history in computational lexical semantics. Given a set of word pairs and corresponding scores quantifying how strongly human assessors deem the two respective words in a word pair semantically related, the goal is automatically produce similar assessment scores. The evaluation is normally carried out in terms of correlation coefficients.

	Count (word2vec)	**Count (GloVe)**
No. of languages with ≥ 50000 words	11	11
No. of languages with ≥ 10000 words	38	38
No. of languages with ≥ 5000 words	62	62
No. of languages with ≥ 1000 words	123	123
No. of languages with ≥ 100 words	267	266
No. of languages with ≥ 10 words	360	360

Table 3: Number of languages.

UKP30	Chandar A P et al. (2014) En-De Vectors	0.212 @ 34.5%
	Ours (word2vec)	0.752 @ 96.6%
	Ours (GloVe)	0.777 @ 96.6%
GUR65	Chandar A P et al. (2014) En-De Vectors	−0.319 @ 26.2%
	Faruqui et al. (2015)	0.603 @ N/A
	Ours (word2vec)	0.717 @ 96.9%
	Ours (GloVe)	0.768 @ 96.9%
GUR350	Chandar A P et al. (2014) En-De Vectors	0.558 @ 51.7%
	Ours (word2vec)	0.605 @ 68.3%
	Ours (GloVe)	0.680 @ 68.0%
ZG222	Chandar A P et al. (2014) En-De Vectors	0.111 @ 38.3%
	Ours (word2vec)	0.161 @ 54.1%
	Ours (GloVe)	0.306 @ 54.1%

Table 4: German semantic relatedness results, evaluated in terms of Spearman's rank correlation coefficient and coverage.

RG65	Chandar A P et al. (2014) En-Es Vectors	0.629 @ 55.4%
	Ours (word2vec)	0.805 @ 100.0%
	Ours (GloVe)	0.844 @ 100.0%
MC30	Chandar A P et al. (2014) En-Es Vectors	0.430 @ 60.0%
	Faruqui et al. (2015)	0.591 @ N/A
	Ours (word2vec)	0.830 @ 76.7%
	Ours (GloVe)	0.853 @ 76.7%
WS353	Chandar A P et al. (2014) En-Es Vectors	0.256 @ 65.1%
	Ours (word2vec)	0.538 @ 65.6%
	Ours (GloVe)	0.596 @ 65.6%

Table 5: Spanish semantic relatedness results, evaluated in terms of Spearman's rank correlation coefficient and coverage.

JI65	Chandar A P et al. (2014) En-Fr Vectors	0.586 @ 49.2%
	Faruqui et al. (2015)	0.606 @ N/A
	Ours (word2vec)	0.822 @ 96.9%
	Ours (GloVe)	0.827 @ 96.9%

Table 6: French semantic relatedness results, evaluated in terms of Spearman's rank correlation coefficient and coverage.

English–German RG65	Chandar A P et al. (2014) En-De Vectors	0.441 @ 38.4%
	Ours (word2vec)	0.812 @ 97.6%
	Ours (GloVe)	0.828 @ 97.6%
English–Spanish RG65	Chandar A P et al. (2014) En-Es Vectors	0.588 @ 59.5%
	Ours (word2vec)	0.869 @ 100.0%
	Ours (GloVe)	0.863 @ 100.0%
English–French RG65	Chandar A P et al. (2014) En-Fr Vectors	0.598 @ 52.0%
	Ours (word2vec)	0.864 @ 100.0%
	Ours (GloVe)	0.855 @ 100.0%
English–Spanish MC30	Chandar A P et al. (2014) En-Es Vectors	0.351 @ 70.0%
	Ours (word2vec)	0.745 @ 90.0%
	Ours (GloVe)	0.797 @ 90.0%
Spanish–English MC30	Chandar A P et al. (2014) En-Es Vectors	0.645 @ 56.7%
	Ours (word2vec)	0.713 @ 83.3%
	Ours (GloVe)	0.721 @ 83.3%
English–Spanish WS353	Chandar A P et al. (2014) En-Es Vectors	0.303 @ 75.9%
	Ours (word2vec)	0.582 @ 79.8%
	Ours (GloVe)	0.641 @ 79.8%
Spanish–English WS353	Chandar A P et al. (2014) En-Es Vectors	0.299 @ 73.3%
	Ours (word2vec)	0.550 @ 78.7%
	Ours (GloVe)	0.612 @ 78.7%

Table 7: Cross-lingual semantic relatedness results

These quantify to what degree the word pairs turn out to be in a similar order when sorting with respect to the two kinds of relatedness scores – ground-truth human-provided ones vs. automatically generated ones.

Although semantic relatedness assessment is not an end-user task in itself, it is an important building block in numerous NLP systems. For instance, measures of semantic relatedness can be used in search query expansion, text classification, and schema and ontology matching, among many others.

Following Pennington et al. (2014), we use cosine similarity over normalized vectors. The word vectors we generate are case-sensitive, distinguishing *Reading*, which often refers to the city, from *reading*, which often refers to the process of reading. However, some of the datasets do not preserve case and so we also consider any possible capitalized version of the input word. Whenever at least one word has multiple candidate vectors, we take the maximum similarity over all pairs.

We evaluate this method using Spearman's rank correlation coefficients over all covered word pairs, while reporting the respective coverage percentage. In computing Spearman's ρ, we follow the recommended procedure of using average ranks for tied positions.

While most semantic relatedness datasets focus on English, there are a few non-English ones as well, which we can use to evaluate our system. Unfortunately, their number is rather small, so we are limited in the number of languages that we can readily consider in this sort of evaluation. We use several publicly released datasets that were often based on pre-existing English datasets[2]. In Table 4 we provide evaluation results on German-language datasets, while Tables 5 and 6 provide similar results on Spanish and French datasets. For comparison, we list all published results known to us that are also based on vectors as well as results on all other non-English word vectors we could obtain and evaluate directly. In all cases, we see that our vectors fare significantly better than the competitors.

4.4 Cross-Lingual Semantic Relatedness

Semantic relatedness can also be evaluated across languages. We adopt the same methodology as earlier but rely on the Spanish-English evaluation data from Hassan and Mihalcea (2009), which we can use to compare our vectors with those of Chandar A P et al. (2014). Further, we consider the new cross-lingual semantic relatedness evaluation data released by Camacho-Collados et al. (2015), which is based on the RG65 dataset. Our results on these cross-lingual datasets are listed in Table 7. Again, our Wiktionary-based representations compare favorably with other available results.

4.5 Word Choice Problems

Word choice problems consist of a target word and a selection of possible words or phrases describing it. Consider the following three examples.

gourmet	**dale**	**brace**
a) enjoys cooking	a) plain	a) to scream
b) has indigestion	b) retreat	b) prepare for danger
c) has an expert appreciation of food	c) shelter	c) hold your breath
d) is hungry	d) valley	d) close your eyes

Here, the correct answers are c) for *gourmet*, d) for *dale*, and b) for *brace*. For English, we rely on a well-known dataset used by Mohammad et al. (2007). We also use a large German-language collection of similar quiz questions[3]. The latter consists of 984 problem instances collected from 2001 to 2005 editions of the German version of Reader's Digest Magazine, where they appear as "Word Power" problems.

We compute cosine similarities between the target word and the candidate answers. Some answers are individual words or expressions already covered in our data, in which case this is simple. If a candidate answer, however, consists of multiple words that are not covered in our data as a multi-word expression, we simply use the maximum cosine similarity between any of the words in the answer phrase and the target word.

We assess the accuracy as the sum of scores over all problem instances divided by the number of problem instances. Following the convention from previous work (Mohammad et al., 2007), the score is 1 if the correct answer is ranked highest among the candidates, 0 if it is not

[2] For more information on these datasets, please refer to `https://www.ukp.tu-darmstadt.de/data/semantic-relatedness/german-relatedness-datasets/` as well as Hassan and Mihalcea (2009) and Camacho-Collados et al. (2015).

[3] `https://www.ukp.tu-darmstadt.de/data/semantic-relatedness/german-word-choice-problems/`

Dataset	Vectors	Accuracy
English	Chandar A P et al. (2014) En-De Vectors	27.42%
	Original word2vec input vectors	65.31%
	Ours (word2vec)	76.49%
	Original GloVe input vectors	68.54%
	Ours (GloVe)	74.42%
German	Chandar A P et al. (2014) En-De Vectors	27.35%
	Ours (word2vec)	40.91%
	Ours (GloVe)	40.85%

Table 8: Accuracy results on English and German word choice problems

ranked highest, and $\frac{1}{n}$ if our method's top ranked answers form a tie of n answers with the same similarity score.

The results are provided in Table 8. Although our method for handling phrases is very simplistic, we obtain reasonably good results. Somewhat surprisingly, we quite significantly improve over the original input vectors for English. The contribution could come from the English lexical information in Wiktionary as well as from the cross-lingual relationships extracted from Wiktionary.

For German, the results are not as good as for English, which, however, is mainly due to the morphological complexity of the phrases in German. Better results could easily be obtained by improving the linguistic analysis of candidate answers, for instance by performing lemmatization, stop word removal or interpretation, and compound splitting, which, of course, is particularly helpful for German with its notoriously long compound nouns. After that, one could then use our vectors to obtain more reliable similarity scores.

4.6 Word and Entity Analogies

Mikolov et al. (2013c) showed that distributed word vectors trained on large corpora using prediction approaches may exhibit intriguing semantic and linguistic regularities, even if these are not in any way directly part of their training objective. For instance, in their results, the vectors for *king* and *queen* stand roughly in the same relationship to each other as the vectors for *uncle* and *aunt*, or *man* and *woman*. This works to the extent that simple vector arithmetic often produces a vector whose nearest known word vector is the correct answer.

In order to create a non-English analogy dataset, we took the semantic analogies dataset of Mikolov et al. (2013c) and filtered out the parts focusing on geography, as these are to a large extent language-independent names like *Portland* or *Alaska*. This left us with the family and male/female related analogies. From these, we randomly selected 50 examples and created the corresponding French-language analogies. When multiple different translations appeared reasonable, we first restricted the choice based on the register (*maman* for *mom* but *mère* for *mother*) and then used the most popular form in the few cases where more than one option remained, e.g. *belle-mère* for *stepmother* rather than *marâtre*, which typically has the connotation of implying an evil stepmother.

For each analogy entry, the first two words demonstrate the analogy, and for the second pair of words, only the first is given as input to the system. The goal is predict the second one. We follow Mikolov et al. (2013c) in computing the target vector using simple vector arithmetic. We then find words near that target vector by choosing the nearest neighbors in terms of the Euclidean distance, considering only French words so as to obtain an answer in the correct target language. On our dataset of 50 French analogies, we obtain the results shown in

Vectors	Accuracy
Chandar A P et al. (2014) En-Fr Vectors	2.0%
Ours (word2vec)	30.0%
Ours (GloVe)	35.0%

Table 9: Accuracy results on the French word analogy task

Table 9. Note that the French dataset is somewhat more challenging than the English original, because some translations are polysemous and no longer retain the sense distinctions of the English originals. For instance, both *girl* and *daughter* correspond to *fille* in French. While the approach by Chandar A P et al. (2014) does extremely poorly, our vectors achieve reasonable results. In those cases where they return the wrong answer, the correct one is often among the top 3.

5 Background and Related Work

Distributed representations in neural networks go back to at least Rumelhart et al. (1986), who described, for their well-known family tree case study, how weights distributed across different input units can be used to describe people. Importantly, their representation allowed two different people from separate families to share most weights if their other attributes were similar.

The lineage of the distinct idea of *distributional semantics* can be traced to use theories of linguistic meaning, which, roughly speaking, hold that language use in context determines the meaning of a word. This view fits well with the idea of empiricist corpus linguistics and the computational goals of discerning aspects of meaning using data-driven methods. Thus, distributional methods have received considerable attention in natural language processing (Schütze, 1993). Over time, it became apparent that one of the challenges with many distributional methods is the sparsity of observed word co-occurrences in a corpus in comparison with the overall distribution of likely word co-occurrences. Since many distributional approaches use numerical vectors to represent the contexts, this sparsity often is manifested in the form of sparse vectors with many zeros. Smoothing techniques and algorithms such as Latent Semantic Analysis (Deerwester et al., 1990) were proposed to alleviate some of these problems.

More recently, distributed and distributional methods have grown together in the form of neural network-inspired architectures that learn distributed representations from large corpora by accounting for word co-occurrences (Collobert et al., 2011; Turian et al., 2010). The resulting representations are still vectorial and based on corpus co-occurrences, but much lower-dimensional than in traditional distributional approaches and thus significantly less sparse. The massive attention on deep learning in recent years, paired with fast training methods as in the word2vec method by Mikolov et al. (2013a), which actually forgoes deep learning, has propelled these methods to the forefront of NLP, to the point that they are known well beyond the core natural language processing community.

Subsequently, a number of improvements to the learning algorithms have been proposed. Our objective function is related to those of other models that aim to exploit similarities between words. Chen et al. (2015) extend the word2vec CBOW objective function in order to pay special attention to contexts that reveal more explicit semantic relationships, rather than treating all contexts as equal. The semantic relationships are obtained using information extraction methods, e.g. from lists and definitions. Yu and Dredze (2014) and Faruqui et al. (2015) propose to optimize monolingual word embeddings so as to match information from pre-existing lexical resources. Hill et al. (2015) used dictionary glosses from several resources (including Wiktionary) in order to train neural networks to produce vectors for multi-word phrases.

While most word representation learning research has been monolingual aiming at English, recently there has been some interest in multilingual aspects of it.

Some works take pre-existing vectors for different languages and connect them. Mikolov et al. (2013b) develop a method to learn projections between two monolingual word embedding spaces. Lazaridou et al. (2015) investigate means to improve such projections. Faruqui and Dyer (2014) propose using canonical correlation analysis (CCA), while Lu et al. (2015) suggest using Deep CCA instead. Our method, in contrast, does not assume that we have already created non-English word vectors. We only rely on English word vectors, which are readily available from numerous sources.

A number of studies have focused on using multilingual corpora, often parallel corpora, to produce bilingual word vector spaces (Kalchbrenner and Blunsom, 2013; Kočiský et al., 2014). Utt and Padó (2014) investigate using syntax for bilingual vector space models. Hermann and Blunsom (2014) create bilingual word representations without word alignment. Hill et al. (2014) showed that word embeddings obtained from translations better reflect the ontological status of words than regular neural embeddings. One advantage of corpus-based approaches is the potential to have a substantial coverage, given sufficiently large corpora. However, as the amount of available parallel text is somewhat limited, in practice, this advantage may only apply to methods that do not require parallel corpora. Our work is complementary to this line of research on corpus-driven approaches. We exploit the availability of high-quality word vectors for English trained on very large Web-scale data, leading to word vector spaces that reflect word analogies well. We further draw on the availability of multilingual lexical resources such as Wiktionary, covering hundreds of languages, including lesser-resourced ones, for which corpora may be difficult to obtain.

More generally, our work differs from previous work by going beyond bilingual vector spaces in order to place millions of word forms from different languages into a single shared vector space.

6 Conclusion

We have presented the first study to produce large amounts of word vectors from Wiktionary in many languages. Unlike previous work on bilingual word embedding spaces, our approach produces a single significantly multilingual word vector space rather than just bilingual ones. Our experiments show that our vectors reflect semantic properties and that they are useful both in monolingual and in cross-lingual settings.

In the future, we would like to investigate the potential of these multilingual vectors for machine translation of text. While deep recurrent neural network architectures have recently achieved state-of-the-art results in several machine translation settings, they still suffer from significant problems with out-of-vocabulary items (Sutskever et al., 2014; Luong et al., 2015). Rather than only addressing these with custom ad hoc techniques as in the approach taken by Luong et al. (2015), it would be helpful to investigate to what extent we can incorporate them within the same vector space.

References

Borin, L., Allwood, J., and de Melo, G. (2014). Bring vs. MTRoget: Evaluating automatic thesaurus translation. In *Proceedings of the 9th Language Resources and Evaluation Conference (LREC 2014)*, Paris, France. ELRA.

Camacho-Collados, J., Pilehvar, M. T., and Navigli, R. (2015). A framework for the construction of monolingual and cross-lingual word similarity datasets. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*.

Chandar A P, S., Lauly, S., Larochelle, H., Khapra, M., Ravindran, B., Raykar, V. C., and Saha, A. (2014). An autoencoder approach to learning bilingual word representations. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pages 1853–1861. Curran Associates, Inc.

Chen, J., Tandon, N., and Gerard de Melo (2015). Neural word representations from large-scale commonsense knowledge. In *Proceedings of the IEEE/WIC/ACM Web Intelligence Conference and the IEEE/WIC/ACM Intelligent Agent Technology Conference 2015 (WI/IAT 2015)*.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

de Melo, G. (2014). Etymological Wordnet: Tracing the history of words. In *Proceedings of the 9th Language Resources and Evaluation Conference (LREC 2014)*, Paris, France. ELRA.

de Melo, G. (2015). Lexvo.org: Language-related information for the linguistic linked data cloud. *Semantic Web*, 6(4).

de Melo, G. and Weikum, G. (2009). Towards a Universal Wordnet by learning from combined evidence. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009)*.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Etzioni, O., Reiter, K., Soderland, S., and Sammer, M. (2007). Lexical translation with application to image search on the web. In *Proceedings of Machine Translation Summit XI*.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2015). Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2015)*.

Faruqui, M. and Dyer, C. (2014). Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*.

Göhring, A. (2014). Building a spanish-german dictionary for hybrid mt. In *Proceedings of the 3rd Workshop on Hybrid Approaches to Translation (HyTra) at EACL 2014*, pages 30–35. Association for Computational Linguistics.

Hassan, S. and Mihalcea, R. (2009). Cross-lingual semantic relatedness using encyclopedic knowledge. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009)*, Singapore.

Hermann, K. M. and Blunsom, P. (2014). Multilingual Distributed Representations without Word Alignment. In *Proceedings of the International Conference on Learning Representations (ICLR) 2014*.

Hill, F., Cho, K., Jean, S., Devin, C., and Bengio, Y. (2014). Not all neural embeddings are born equal. *CoRR – Computing Research Repository – arXiv*, abs/1410.0718.

Hill, F., Cho, K., Korhonen, A., and Bengio, Y. (2015). Learning to understand phrases by embedding the dictionary. *CoRR – Computing Research Repository – arXiv*, abs/1504.00548.

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*, Seattle. Association for Computational Linguistics.

Kočiský, T., Hermann, K. M., and Blunsom, P. (2014). Learning Bilingual Word Representations by Marginalizing Alignments. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), Volume 2: Short Papers*.

Lazaridou, A., Dinu, G., and Baroni, M. (2015). Hubness and pollution: Delving into cross-space mapping for zero-shot learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*.

Li, S., Graça, J. a. V., and Taskar, B. (2012). Wiki-ly supervised part-of-speech tagging. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1389–1398, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lu, A., Wang, W., Bansal, M., Gimpel, K., and Livescu, K. (2015). Deep multilingual correlation for improved word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2015)*, pages 250–256, Denver, Colorado. Association for Computational Linguistics.

Luong, T., Sutskever, I., Le, Q., Vinyals, O., and Zaremba, W. (2015). Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *CoRR – Computing Research Repository – arXiv*, abs/1301.3781.

Mikolov, T., Le, Q. V., and Sutskever, I. (2013b). Exploiting similarities among languages for machine translation. *CoRR – Computing Research Repository – arXiv*, abs/1309.4168.

Mikolov, T., tau Yih, W., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*. Association for Computational Linguistics.

Mohammad, S., Gurevych, I., Hirst, G., and Zesch, T. (2007). Cross-lingual distributional profiles of concepts for measuring semantic distance. In *2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*.

Nastase, V. and Strapparava, C. (2013). Bridging languages through etymology: The case of cross language text categorization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 651–659. The Association for Computer Linguistics.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1532–1543.

Richman, A. and Schone, P. (2008). Mining Wiki Resources for Multilingual Named Entity Recognition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, pages 1–9.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323:533–536.

Schütze, H. (1993). Word space. In *Advances in Neural Information Processing Systems 5 (NIPS 1992)*.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, Montreal, CA.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*, pages 384–394.

Utt, J. and Padó, S. (2014). Crosslingual and multilingual construction of syntax-based vector space models. *Transactions of the Association of Computational Linguistics*, 2:245–258.

Yu, M. and Dredze, M. (2014). Improving lexical embeddings with semantic knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), Volume 2: Short Papers*.