Wikifying novel words to mixtures of Wikipedia senses by structured sparse coding*

Balázs Pintér, Gyula Vörös, Zoltán Szabó, and András Lőrincz

Faculty of Informatics, Eötvös Loránd University,
Pázmány P. sétány 1/C, H-1117 Budapest, Hungary
bli@elte.hu,vorosgy@inf.elte.hu,szzoli@cs.elte.hu,andras.lorincz@elte.hu

Abstract. We extend the scope of Wikification to novel words by relaxing two premises of Wikification: (i) we wikify without using the surface form of the word (ii) to a mixture of Wikipedia senses instead of a single sense. We identify two types of "novel" words: words where the connection between their surface form and their meaning is broken (e.g., a misspelled word), and words where there is no meaning to connect to – the meaning itself is also novel.

We propose a method capable of wikifying both types of novel words while also dealing with the inherently large-scale disambiguation problem. We show that the method can disambiguate between up to 1000 Wikipedia senses, and it can explain words with novel meaning as a mixture of other, possibly related senses. This mixture representation compares favorably to the widely used bag of words representation.

Keywords: interpreting novel words, Wikification, link disambiguation, natural language processing, structured sparse coding

1 Introduction

Wikification aims to help users and computers alike in understanding texts by enriching them with encyclopedic knowledge in the form of links to Wikipedia articles [1]. However, Wikification concerns itself only with correct and known words: neologisms, misspelled words and the like fall outside its scope.

These *novel words* are different in that the connection between their surface form[1] and their meaning is broken (e.g., a misspelled word), or – in the more involved case – there is no meaning to connect to (e.g., a word with a completely new meaning). This property makes them particularly hard to interpret, but it also makes them the words that need interpreting the most.

This paper extends the scope of Wikification to *novel words* by interpreting them (i) *without relying on their surface form* and (ii) as a *weighted mixture of Wikipedia senses*, instead of as a single sense.

[*] In Janusz Kacprzyk, editor, Advances in Intelligent and Soft Computing. © Springer-Verlag Berlin Heidelberg 2013. The original publication is available at www.springerlink.com.

[1] the form of a word as it appears in the text

Usually, Wikification consists of two phases: *link detection* and *link disambiguation*. The detection phase identifies the terms and phrases from which links should be made. The disambiguation phase identifies the appropriate Wikipedia article for each detected term to link to. For example, the term *bank* could link to an article about financial institutions or river banks. We consider only disambiguation, as the words to be disambiguated are assumed given: they are the novel words in the text.

Similarly to Mihalcea [1], we regard Wikipedia as a *sense inventory*, where each link can be thought of as a sense-annotated word. In each link, the anchor text of the link – the word – is annotated with the target Wikipedia page – the sense.

Novel words can be of two types with respect to this sense inventory. In the first case, a novel surface form is – maybe incorrectly – associated with an already known meaning. An example for correct word use is a neologism where a new word gets associated with an already known sense (e.g., neologisms created by clipping: professor → prof, facsimile → fax). Examples for words used incorrectly include misspelled words, mixed up words like homophones, scanning or Optical Character Recognition errors, errors introduced by automatic speech recognition, etc. For the sake of simplicity, we also refer to these as *novel words*, although they may be completely unintelligible (e.g., a word completely blurred in a scanned document).

In the second case, the meaning of the novel word itself is also novel – it is not present in the sense inventory. In many cases, these words can be explained by a mixture of senses. A striking example is neologisms created by blending, like edutainment (from education and entertainment) and netiquette (from network and etiquette) [2]. Even in less clear-cut cases, finding a set of senses closely related to the novel meaning could help users and computer algorithms alike to understand it.

To interpret these novel words, we have to overcome a new difficulty. As we do not rely on the surface form of the target word[2], the *complexity of the disambiguation problem increases*. Current methods for Wikification treat the disambiguation of different word types[3] independently. In the case of novel words, we cannot formulate an independent problem for each surface form; we have to disambiguate among hundreds or thousands of senses at once instead of about a dozen. This vast number of candidate senses results in a large-scale problem, and this is why the new difficulty appears.

Typical methods to disambiguate words with *correct* surface form apply the *distributional hypothesis*. According to the distributional hypothesis, words that occur in the same contexts tend to have similar meanings [3]. Because our new disambiguation problem without using the surface form is large-scale, exceptions to the distributional hypothesis occur more frequently. Particularly, let us call two contexts *spuriously similar* if they are similar but belong to words that de-

[2] the word to be explained with Wikipedia senses

[3] In “A rose is a rose is a rose”, there are three word types (a, rose, is), but eight word tokens.

note different senses. The number of spuriously similar contexts tends to increase inherently with the number of candidate senses. There is more chance to select a wrong sense from among 1000 senses than from among 10: the learning problem becomes considerably harder.

To counter the effect of spurious similarities, we use the *distributional hypothesis* in a novel way. We introduce structured sparse coding [4] to diminish the effect of spurious similarities of contexts by matching the structure in the regularization to the structure of the problem (Section 3).

The **contributions** of the paper are summarized as follows: (i) we propose a method to interpret novel words as weighted mixtures of Wikipedia senses. (ii) We show that structured sparsity reduces the effect of spurious similarities of contexts. (iii) We perform large-scale evaluations where we disambiguate among 1000 Wikipedia senses at once.

In the next section we review related work. Our method and results are described in Section 3 and 4. We discuss our results in Section 5 and conclude in Section 6.

2 Related Work

The main differences between previous methods for Wikification and ours is that they consider the disambiguation problems of different word types independently, and they wikify to a single Wikipedia sense. We relaxed these two premises to make interpreting novel words possible.

Mihalcea et al. [1] introduced the concept of Wikification: they proposed a method to automatically enrich text with links to Wikipedia articles. They used keyword extraction to detect the most important terms in the text, and disambiguated them to Wikipedia articles with supervised learning using the contexts. The same task was solved in [5] more efficiently. Here, contexts were taken into account also for the detection phase. Disambiguation was done using sense *commonness* and sense *relatedness* scores.

Unlike the previously mentioned works, which introduce links to important terms in the text chiefly to achieve better readability, the goal of [6] was to add as many links as possible to help information retrieval. The terms were disambiguated by assuming that coherent documents refer to entities from one or a few related topics or domains. Ratinov et al. [7] proposed a similar disambiguation system called Glow (Global Wikification), which used several local and global features to obtain a set of disambiguations that are coherent in the whole text.

In information retrieval and speech recognition, unintelligible words pose a practical problem. The TREC-5 confusion track [8] studied the impact of data corruption introduced by scanning or Optical Character Recognition errors on retrieval performance. In the subsequent spoken document retrieval tracks [9], the errors were introduced by automatic speech recognition.

Structured sparsity has been successfully applied to natural language processing problems, e.g., in [10] and [11]. Jenatton et al. [10] apply sparse hierarchical dictionary learning to learn hierarchies of topics from a corpora of NIPS proceedings papers. In a more recent application [11], structured sparsity was used to perform effective feature template selection on three natural language processing tasks: chunking, entity recognition, and dependency parsing.

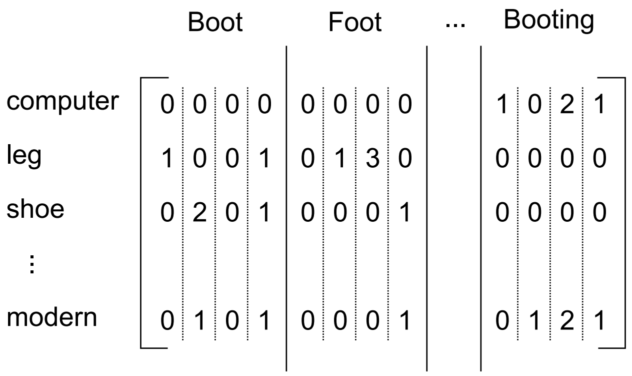
3 The Method

The novel word is explained as a weighted mixture of Wikipedia senses. Particularly, we assign a vector of coefficients to each novel word – an *interpretation vector* – where each coefficient corresponds to a single Wikipedia sense.

The interpretation vector is determined in two steps. First, we formulate a linear model with a structured sparsity inducing regularization and compute a representation vector $\boldsymbol{\alpha}$. In the second step, this representation vector is condensed to yield an interpretation vector.

We start with a set of Wikipedia senses the novel word could be interpreted as. For each *sense*, we collect a number of *contexts* from Wikipedia. A context of a sense consists of the N non-stopword words occurring before and after the anchor of the link that points to the corresponding Wikipedia page. For example, the anchor text `bar` could point to (and be tagged with) `Bar_(law)`, `Bar_(unit)`, `Bar_(establishment)`, etc. There can be at most $2N$ words in a context.

The presented method makes use of a collection of such *contexts* arranged in a word-context matrix $\mathbf{D}$ [12] (Figure 1). In this matrix, each context is a column represented as a bag-of-words vector $\mathbf{v}$ of word frequencies, where v_i is the number of occurrences of the ith word in the context.



$$\begin{array}{l|cccc|cccc|c|cccc|}
 & \multicolumn{4}{c}{\text{Boot}} & \multicolumn{4}{c}{\text{Foot}} & \cdots & \multicolumn{4}{c}{\text{Booting}} \\
\text{computer} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & 1 & 0 & 2 & 1 \\
\text{leg} & 1 & 0 & 0 & 1 & 0 & 1 & 3 & 0 & & 0 & 0 & 0 & 0 \\
\text{shoe} & 0 & 2 & 0 & 1 & 0 & 0 & 0 & 1 & & 0 & 0 & 0 & 0 \\
\vdots & & & & & & & & & & & & & \\
\text{modern} & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & & 0 & 1 & 2 & 1
\end{array}$$

Fig. 1. The word-context matrix $\mathbf{D}$. Each column is a context of a Wikipedia sense (e.g., *Boot*, *Foot*). Each element D_{ij} of the matrix holds the number of occurences of the ith word in the jth context. For example, the word *leg* occurs three times in the 7th context, which is the 3rd context labeled with *Foot*.

To compute the representation vector $\boldsymbol{\alpha}$, the context $\mathbf{x} \in \mathbb{R}^m$ of the target word is approximated linearly with the columns of the word-context matrix $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n] \in \mathbb{R}^{m \times n}$, called the dictionary in the terminology of sparse coding. The columns of the dictionary contain contexts, each labeled with the sense $l_i \in L$ the context was collected for. Please note that multiple contexts can be, and in many cases are, tagged with the same sense: $l_i = l_j$ is possible. There are m words in the vocabulary, and n contexts in the dictionary.

The representation vector $\boldsymbol{\alpha}$ consists of the coefficients of a linear combination

$$\mathbf{x} \approx \alpha_1 \mathbf{d}_1 + \alpha_2 \mathbf{d}_2 + \ldots + \alpha_n \mathbf{d}_n. \tag{1}$$

For each target word, whose context is $\mathbf{x} \in \mathbb{R}^m$, a representation vector $\boldsymbol{\alpha} = [\alpha_1; \alpha_2; \ldots; \alpha_n] \in \mathbb{R}^n$ is computed.

We introduce the structured sparsity inducing regularization by organizing the contexts in $\mathbf{D}$ into *groups*. Each group contains the contexts annotated with a single sense. Sparsity on the groups is realized by computing $\boldsymbol{\alpha}$ with a *group Lasso* regularization [13] determined by the labels.

The groups are introduced as a family of sets $\mathcal{G} = \{G_l\}_{l \in L} \subseteq 2^{\{1,\ldots,n\}}$. There are as many sets in $\mathcal{G}$ as there are distinct senses in L. For each sense $l \in L$, there is exactly one set $G_l \in \mathcal{G}$ that contains the indices of all the columns $\mathbf{d}_i$ tagged with l. $\mathcal{G}$ forms a partition.

The representation vector $\boldsymbol{\alpha}$ of the target word whose context is $\mathbf{x}$ is computed as the minimum of the loss function

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda \sum_{l \in L} w_l ||\boldsymbol{\alpha}_{G_l}||_2, \tag{2}$$

where $\boldsymbol{\alpha}_{G_l} \in \mathbb{R}^{|G_l|}$ denotes the vector where only the coordinates present in the set $G_l \subseteq \{1, \ldots, n\}$ are retained.

The first term is the approximation error, the second one realizes the structured sparsity inducing regularization. Parameter $\lambda > 0$ controls the tradeoff between the two terms. The parameters $w_l > 0$ denote the weights for each group G_l.

If each group is a singleton (i.e., $\mathcal{G} = \{\{1\}, \{2\}, \ldots, \{n\}\}$) the Lasso problem [14] is recovered:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda \sum_{i=1}^{n} w_i |\alpha_i|. \tag{3}$$

Setting $\lambda = 0$ yields the least squares cost function.

For the sake of simplicity, we represent each sense with the same number of contexts: there are an equal number of columns in $\mathbf{D}$ for each label $l \in L$ ($|G_1| = |G_2| = \cdots = |G_{|L|}|$). The weights w_l of the groups are set to 1.

In the *second step*, the target word is disambiguated to a mixture of Wikipedia senses based on the weights in this vector. We utilize the group structure to condense the vector $\boldsymbol{\alpha} \in \mathbb{R}^n$ to a vector $\mathbf{s} \in \mathbb{R}^{|L|}$ where each coordinate corresponds to a single sense. The interpretation vector is obtained by summing the weights in each group $G_l \in \mathcal{G}$. The weight for each sense $l \in L$ in the mixture is

$$s_l = \sum_i (\boldsymbol{\alpha}_{G_l})_i. \tag{4}$$

The structured sparsity inducing regularization fulfills three purposes. Firstly, it allows us to conveniently condense the representation vector $\boldsymbol{\alpha}$ to the interpretation vector $\mathbf{s}$ based on the groups. Secondly, it allows us to explain each target word with only a few senses. This is important mainly for applications where human users interpret the results.

Thirdly, and most importantly, the structured sparsity inducing regularization allows us to reduce the effect of spurious similarities of contexts in the large-scale disambiguation problem, as it selects *whole groups of contexts*.

Each group $G_l \in \mathcal{G}$ contains contexts tagged with the same sense $l \in L$, and only a few groups can be selected. The 2-norm in the loss function favors dense representations: it tries to represent each selected sense densely in the representation vector $\boldsymbol{\alpha}$. The method tends to choose representations where most of the contexts are active in the group of a selected sense over representations where only a few contexts are active. Intuitively, a context that is similar to the context of the target word only by accident – the context in the group of an incorrect sense – won't be selected, as most of the other contexts in its group will be dissimilar, and so inactive. In the group of the correct sense, most of the contexts will be similar and active, so that will be selected instead.

An important consequence of reducing the effect of spurious similarities is increased accuracy in large-scale problems compared to other algorithms (Sections 4 and 5).

4 Results

We evaluate the proposed method on two tasks for the two types of novel words. In the first task, we use the method to interpret words whose connection between their surface form and their meaning is broken, but the sense they denote is present in our sense inventory. These include misspelled words, certain neologisms, errors introduced by automatic speech recognition, and the like (Section 1).

In the second task, we interpret words with novel meaning. These are words for whom there are no correct senses in our sense inventory. Our expectation is that the meaning of these words can be approximated by mixtures of related senses. We compare the quality of the interpretation vectors to the bag of words contexts by measuring the quality of the clustering they induce.

4.1 The Datasets

The datasets used in our experiments are obtained by randomly sampling the links in Wikipedia. Each dataset consists of contexts tagged with senses $(\mathbf{c}_1, l_1), (\mathbf{c}_2, l_2), \ldots$. Each tagged context is obtained by processing a link: the bag-of-words vector generated from the context of the anchor text is annotated with the target of the link.

We use the English Wikipedia database dump from October 2010[4]. Disambiguation pages, and articles that are too small to be relevant (i.e., have less than 200 non-stopwords in their texts, or less than 20 incoming and 20 outgoing links) are discarded. Inflected words are reduced to root forms by the Porter stemming algorithm [15].

To produce a dataset, a list of anchor texts are generated that match a number of criteria. These criteria have been chosen to obtain (i) words that are frequent enough to be suitable training examples and (ii) are proper English words. The anchor text has to be a single word between 3 and 20 characters long, must consist of the letters of the English alphabet, must be present in Wikipedia at least 100 times, and must point to at least two different Wikipedia pages, but not to more than 20. It has to occur at least once in *WordNet* [16] and at least three times in the *British National Corpus* [17].

A number of anchor texts are selected from this list randomly, and their linked occurrences are collected along with their N-wide contexts. Each link is processed to obtain a labeled context $(\mathbf{c}_i, l_i)$.

To ensure that there are an equal number of contexts tagged with each sense $l \in L$, d randomly selected contexts are collected for each label. Labels with less than d contexts are discarded. We do not perform feature selection, but we remove the words that appear less than five times across all contexts, in order to discard very rare words.

4.2 Interpreting novel words whose meaning is present in the sense inventory

The first task is a disambiguation problem where the algorithm is used to select a *single correct sense* from *all the available senses* in the sense inventory. Given a context $\mathbf{x} \in \mathbb{R}^m$ of a word, the goal is to determine the correct sense $l \in L$. The performance of the algorithms is measured as the accuracy of this classification.

We compare the interpretation vectors computed with group Lasso to three baselines: representations $\boldsymbol{\alpha}$ computed with two different regularizations (least squares and the Lasso) of the linear model described in Section 3, and a Support Vector Machine (SVM). The SVM is a multiclass Support Vector Machine with a linear kernel, used successfully for Wikification in previous works [5, 7].

The interpretation vector $\mathbf{s}$ yields a single sense by simply selecting its largest coefficient. Similarly for least squares and the Lasso, the target word is disambiguated to the sense that corresponds to the largest coefficient in $\boldsymbol{\alpha}$. For the SVM, a classification problem is solved using the labeled contexts $(\mathbf{c}_i, l_i)$ as training and test examples.

The minimization problems of both the *Lasso* and *group Lasso* (Eq. 2) are solved by the Sparse Learning with Efficient Projections (SLEP) package [18]. For the *support vector machine*, we use the implementation of LIBSVM [19].

[4] Downloaded from `http://dumps.wikimedia.org/enwiki/`.

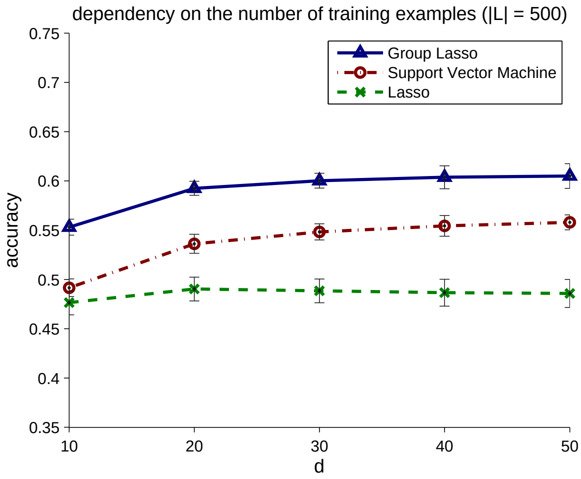


Fig. 2. Dependency of the accuracy on the number of contexts per candidate sense. There are $d - 1$ such contexts in each step of the cross-validation, as there is one test example for each sense. The data points are the mean of values obtained on the five datasets. The error bars denote the standard deviations. "Group Lasso" means taking the largest weight in the interpretation vector computed with group Lasso. The results of least squares are not illustrated as the standard deviations were very large. It performs consistently below the Lasso.

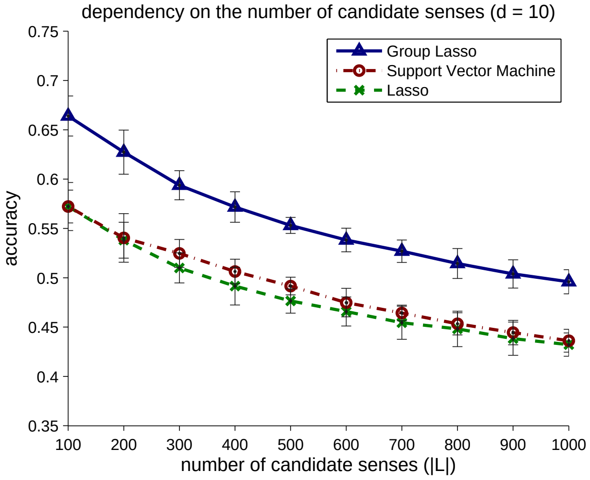
The algorithms are evaluated on five disjoint datasets generated from Wikipedia (Section 4.1), each with different senses. We report the mean and standard deviation of the accuracy across these five datasets.

There are $|L| = 1000$ different senses in each dataset, and $d = 50$ contexts annotated with each sense. The algorithms are evaluated on datasets of different sizes (i.e., d and $|L|$ are different), generated from the original five datasets by removing contexts and their labels randomly.

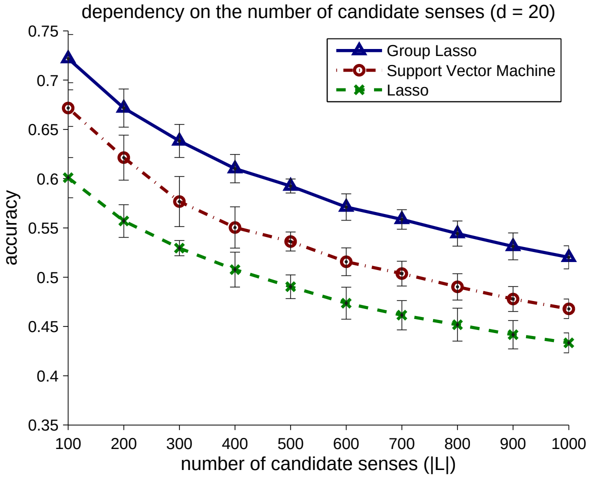
In accord with [20, 21], and others, we use a broad context, $N = 20$. We found that a broad context improves the performance of all four algorithms.

Before evaluating the algorithms, we examined the effect of their parameters on the results. We found that the algorithms are robust: for the Lasso, $\lambda = 0.005$, for the group Lasso, $\lambda = 0.05$, and for the SVM, $C = 1$ was optimal in almost every validation experiment.

In the first evaluation, we examine the effect the number of training examples per candidate sense has on the accuracy of the four algorithms. The starting datasets consist of $|L| = 500$ senses with $d = 10$ contexts (or training examples) each. Stratified 10-fold cross-validation is used to determine the accuracy of the classification: the dataset is partitioned into 10 subsets (the same as d), where each subset contains exactly $|L|$ examples – one annotated with each sense. In one iteration, one subset is used for testing, and the other 9 subsets form the



(a)



(b)

Fig. 3. Dependency of the accuracy on the number of candidate senses, $|L|$. The data points are the mean of values obtained on the five datasets. The error bars denote the standard deviations. "Group Lasso" means taking the largest weight in the interpretation vector computed with group Lasso. The results of least squares are not illustrated, as the standard deviations were very large. It performs consistently below the Lasso.

columns of $\mathbf{D}$: there are $|L|$ test examples and $n = (d-1)|L|$ columns in $\mathbf{D}$ in each iteration. For the SVM, the columns of $\mathbf{D}$ are used as training examples.

To examine the effect of additional contexts, we add contexts to $\mathbf{D}$ for each candidate sense, and examine the change in accuracy. In order to evaluate the effect correctly (i.e., to not make the learning problem harder), the test examples remain the same as with $d = 10$. In other words, we perform the same cross-validation as before, only we add additional columns to $\mathbf{D}$ in each step. In Figure 2, we report the results for $d = 10, 20, 30, 40, 50$.

In the second evaluation, the accuracy of the algorithms is examined as the number of candidate senses $|L|$ increases. As in the first evaluation, there are $d = 10$ examples per candidate sense, and stratified 10-fold cross-validation is performed. Then, the number of examples is raised to $d = 20$ in the same way (i.e., the new examples are not added to the test examples). We report the results for $|L| = 100, 200, \ldots, 1000$ candidate senses in Figure 3.

4.3 Interpreting words with novel meaning

In this section, we extend our examinations of the presented method to interpret words whose meaning is novel. In practice this means that we remove all knowledge about the senses our target words denote from the dictionary $\mathbf{D}$. The word with novel meaning has to be interpreted based on its relatedness to other, possibly related senses.

Words with novel meaning are simulated by making sure that there is no context in the dictionary tagged with any sense the test examples are tagged with. Wikipedia senses in the set T and the contexts tagged by them constitute the test examples (i.e., the contexts of words with novel meaning), while the rest of the senses in L together with their contexts form $\mathbf{D}$. The sets T and L, and so the examples for the words with novel meaning and $\mathbf{D}$ are disjoint: there is not a single context in $\mathbf{D}$ for any of the senses in T.

The evaluation is based on the labeling of the test examples: for each target word, we already know the sense it denotes. This labeling determines a *clustering* of the resulting interpretation vectors $\mathbf{s} \in \mathbb{R}^{|L|}$: two interpretation vectors belong to the same cluster if and only if they are tagged with the same sense. The quality of the interpretation vectors (the performance of the presented method) is measured as the quality of this clustering.

Clustering quality can be measured by various clustering validation measures [22]. For our purposes, we need to consider different criteria than Liu et al. [22], as we do not evaluate the clustering, but the data. Our measure should be able to compare data in coordinate spaces of different dimension, and it should be somewhat sensitive to noise and clusters of different density. On the other hand, the capability to accurately tell the number of clusters in the dataset is not important for us. Based on these criteria, we chose the well-known *R-squared* measure. R-squared may be considered a measure of the degree of difference between clusters and the degree of homogeneity between groups [23, 24].

If X denotes all the test examples, $\mathbf{c}$ is the center of X, $C_t, t \in T$ are the different clusters, and $\mathbf{c}_t$ are the centers of the clusters, then R-squared is

$$RS = \left(\sum_{\mathbf{x} \in X} \|\mathbf{x} - \mathbf{c}\|_2^2 - \sum_{t \in T} \sum_{\mathbf{x} \in C_t} \|\mathbf{x} - \mathbf{c}_t\|_2^2 \right) / \sum_{\mathbf{x} \in X} \|\mathbf{x} - \mathbf{c}\|_2^2. \tag{5}$$

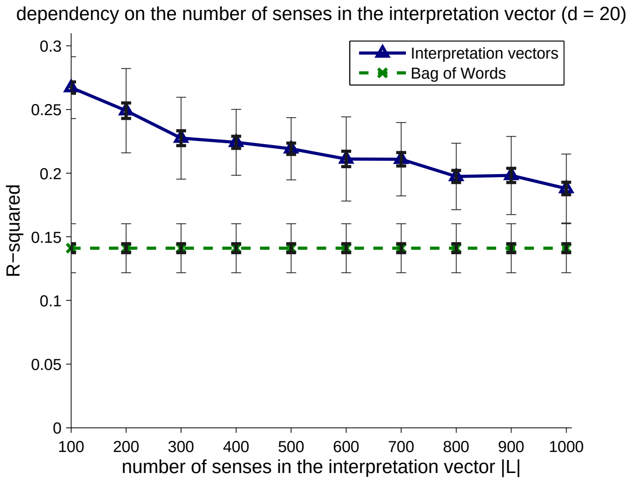


Fig. 4. Interpretation vectors of words with novel meaning vs. the bag of words contexts. The R-squared with the bag of words representation is constant, as it does not depend on the number of senses in the interpretation vector, $|L|$. The data points are the mean of 30 experiments. The thick error bars denote the standard errors of the mean. The thin error bars denote the standard deviations.

For these evaluations, we obtain a single dataset by concatenating the five datasets used in the first task into a larger dataset that contains 5000 senses. The disjoint sets T and L are randomly selected from among these 5000 senses in each experiment.

Parameter λ was set to $\lambda = 0.05$, the same as in the first task. This value yields interpretation vectors with approximately 30 to 70 active senses on average. There are $d = 20$ contexts for each sense. We interpret $|T| = 50$ different senses in each experiment, so there are 1000 target words to interpret. Each experiment is repeated 30 times with different randomly selected senses in both T and L. We report the mean, its standard error, and the standard deviation.

We compare the interpretation vectors to the input bag of words contexts. For each sense $t \in T$, we use the same $d = 20$ contexts that were transformed into the interpretation vectors. For bag of words, we conducted a single set of experiments, as the results do not depend on the value of the parameter $|L|$. We report the results in Figure 4.

5 Discussion

In the first task, the results are very consistent across the five disjoint datasets, except in the case when the representation vector was computed with least squares. The performance of least squares was the worst of the four algorithms, and it was so erratic that we did not plot it in order to keep the figure uncluttered.

For group Lasso and the SVM, additional training examples help up to 20 examples per sense (Figure 2), but only small gains can be achieved by adding more than 20 examples.

The Lasso-based representation does not benefit from new training examples *at all* when there are many candidate senses. This may be the effect of spurious similarities. As more and more contexts are added, the less chance Lasso has to select the right sense from among the candidates.

Classification based on interpretation vectors computed with group Lasso significantly outperforms the other methods, including SVM (Figure 3). This illustrates the efficiency of our method: structured sparsity decreases the chance of selecting contexts spuriously similar to the context of the target word.

In the second task, we found that even when the correct sense of the novel word is unknown, the interpretation vectors perform much better than the bag of words contexts. This points to the possibility of improving performance in natural language processing tasks by using interpretation vectors instead of a bag of words representation.

As the number of senses in the interpretation vector increases, the learning problem becomes harder, and the performance decreases – similarly to the first task. Although there are more and more senses to represent meaning with, these senses were selected randomly from Wikipedia: the chance for senses that are closely related to the novel meaning to appear is too low to offset the effect of the harder learning problem. Based on this intuition, we believe that there is a promising direction for future improvement of the method.

In these first experiments, we interpreted words with novel meaning as mixtures of senses that were randomly selected from Wikipedia. Our experience suggests that a promising avenue of future research is to preselect the senses systematically based on the context of the target word to increase the chance of closely related senses to appear. We have observed some interesting examples where the (unavailable) novel meaning was represented by a mixture of closely related senses. For example, for the novel meaning `Prime_number`, its hypernym, `Number` was selected. For `Existence`, the method selected `Logos`, `Karma`, and `Eternity`. The most interesting example is that of `Transformers`: it was interpreted as a mixture of `Humanoid`, `Tram`, `Flash_(comics)`, `Cyborg`, and `Hero`. With a slight stretch of the imagination, `Transformers` are `Humanoid` robots (`Cyborg`) that can change into vehicles (`Tram`), and they are also `Heroes` that appear in comic books (`Flash_(comics)`) and animated series.

6 Conclusions

We extended the scope of Wikification to novel words by relaxing its premises: (i) we wikify without using the surface form of the word (ii) to a mixture of Wikipedia senses instead of a single sense.

We identified two types of novel words: words where the connection between their surface form and their meaning is broken, and words where there is no meaning to connect to – the meaning itself is also novel.

We proposed a method capable of wikifying both types of novel words while also dealing with the problem of spuriously similar contexts that intensifies because the disambiguation problem becomes inherently large-scale. The performance of the method was demonstrated on two tasks for the two types of novel words. We found that the method was capable of disambiguating between up to 1000 Wikipedia senses. Additionally, we used it to explain words with novel meaning as a mixture of other, possibly related senses. This mixture representation compared favorably to the bag of words input contexts.

In these first experiments of interpreting words with novel meaning, the sense inventories were randomly generated from Wikipedia. Our experience suggests that extending the method by constructing the sense inventory in a systematic way based on the context of the target word is a promising direction for future research.

A possible future application of the presented method is the verification of links to Wikipedia. The method assigns a weight to each candidate sense. If the weight corresponding to the target of the link is small in contrast to weights of other pages, the link is probably incorrect.

The method can be generalized, as it can work with arbitrarily labeled text fragments as well as contexts of Wikipedia links. This more general framework may have further applications, as the idea of distributional similarity offers solutions to many natural language processing problems. For example, topics might be assigned to documents as in centroid-based document classification [25].

ACKNOWLEDGEMENTS

The research has been supported by the 'European Robotic Surgery' EC FP7 grant (no.: 288233). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of other members of the consortium or the European Commission.

The research was carried out as part of the EITKIC_12-1-2012-0001 project, which is supported by the Hungarian Government, managed by the National Development Agency, financed by the Research and Technology Innovation Fund and was performed in cooperation with the EIT ICT Labs Budapest Associate Partner Group.

References

1. Mihalcea, R., Csomai, A.: Wikify!: linking documents to encyclopedic knowledge. In: Proceedings of the Conference on Information and Knowledge Management (CIKM). (2007) 233–242
2. Akmajian, A.: Linguistics: An introduction to language and communication. The MIT press (2001)

3. Harris, Z.: Distributional structure. Word **10**(23) (1954) 146–162
4. Bach, F., Jenatton, R., Mairal, J., Obozinski, G.: Optimization with sparsity-inducing penalties. Foundations and Trends in Machine Learning **4**(1) (2012) 1–106
5. Milne, D., Witten, I.H.: Learning to link with Wikipedia. In: Proceedings of the Conference on Information and Knowledge Management (CIKM). (2008) 509–518
6. Kulkarni, S., Singh, A., Ramakrishnan, G., Chakrabarti, S.: Collective annotation of Wikipedia entities in web text. In: Proceedings of the Conference on Knowledge Discovery and Data Mining (KDD). (2009) 457–466
7. Ratinov, L., Roth, D., Downey, D., Anderson, M.: Local and global algorithms for disambiguation to Wikipedia. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. (2011) 1375–1384
8. Kantor, P.B., Voorhees, E.M.: The TREC-5 Confusion Track: Comparing Retrieval Methods for Scanned Text. Inform. Retrieval **2** (2000) 165–176
9. Garofolo, J.S., Auzanne, C.G.P., Voorhees, E.M.: The TREC Spoken Document Retrieval Track: A Success Story. In: RIAO. (2000) 1–20
10. Jenatton, R., Mairal, J., Obozinski, G., Bach, F.: Proximal methods for hierarchical sparse coding. J. Mach. Learn. Res. **12** (2011) 2297–2334
11. Martins, A.F.T., Smith, N.A., Aguiar, P.M.Q., Figueiredo, M.A.T.: Structured Sparsity in Structured Prediction. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). (2011) 1500–1511
12. Turney, P.D., Pantel, P.: From frequency to meaning: vector space models of semantics. J. Artif. Intell. Res. **37**(1) (2010) 141–188
13. Yuan, M., Yuan, M., Lin, Y., Lin, Y.: Model selection and estimation in regression with grouped variables. J. R. Stat. Soc. **68** (2006) 49–67
14. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. R. Stat. Soc. **58** (1994) 267–288
15. Porter, M.F. In: An algorithm for suffix stripping. Morgan Kaufmann Publishers Inc. (1997) 313–316
16. Miller, G.A.: WordNet: A lexical database for English. Communications of the ACM **38** (1995) 39–41
17. BNC Consortium: The British National Corpus, version 2 (BNC World) (2001)
18. Liu, J., Ji, S., Ye, J.: SLEP: Sparse Learning with Efficient Projections. Arizona State University. (2009)
19. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001)
20. Lee, Y.K., Ng, H.T.: An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). (2002) 41–48
21. Schütze, H.: Automatic word sense discrimination. Comput. Linguist. **24**(1) (1998) 97–123
22. Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J.: Understanding of internal clustering validation measures. In: IEEE 10th International Conference on Data Mining (ICDM), IEEE (2010) 911–916
23. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. J. Intell. Inf. Syst. **17**(2-3) (2001) 107–145
24. Sharma, S.: Applied multivariate techniques. John Wiley & Sons, Inc., New York, NY, USA (1996)
25. Han, E.H., Karypis, G.: Centroid-based document classification: Analysis and experimental results. In: Proceedings of the Conference on Principles of Data Mining and Knowledge Discovery (PKDD). (2000) 116–123