Robust Text Classification for Sparsely Labelled Data Using Multi-level Embeddings

Simon Baker [1,2] **Douwe Kiela** [1] **Anna Korhonen** [2]
[1]Computer Laboratory, 15 JJ Thomson Avenue
[2] Language Technology Lab, DTAL
University of Cambridge, UK
{sb895|dk427|alk23}@cam.ac.uk

Abstract

The conventional solution for handling sparsely labelled data is extensive feature engineering. This is time consuming and task and domain specific. We present a novel approach for learning embedded features that aims to alleviate this problem. Our approach jointly learns embeddings at different levels of granularity (word, sentence and document) along with the class labels. The intuition is that topic semantics represented by embeddings at multiple levels results in better classification. We evaluate this approach in unsupervised and semi-supervised settings on two sparsely labelled classification tasks, outperforming the handcrafted models and several embedding baselines.

1 Introduction

The objective of text classification is to label a scope of text according to predefined labels. While general domains tend to have sufficient amounts of labelled data, in specialised domains (e.g., scientific literature) such data are often scarce and labelled instances number in the hundreds, or low thousands at most. Such domains may also require highly specialised annotators, making labelled data expensive and difficult to obtain (Simpson and Demner-Fushman, 2012).

In order to mitigate the data sparsity problem, a lot of handcrafting is needed to engineer features specific to the task and domain. Typically this process involves a long NLP pipeline, e.g., POS-tagging, parsing, named entity recognition, semantic role labelling, feature selection, etc. Consequently, approaches based on handcrafting can be prohibitively time consuming, and since the resultant features are domain dependent, these systems are difficult to port to other domains (Sebastiani, 2002; Dai et al., 2007). While unsupervised and lightly-supervised methods can bypass the need for labelled data, they in turn tend to suffer from lower performance (Zhang and Elhadad, 2013; Quan et al., 2014; Aggarwal and Zhai, 2012).

In this paper, we present a novel approach to text classification that is especially beneficial in situations were labelled datasets are small. Our approach builds on the Distributed Memory (DM) model by Le and Mikolov (2014). The fast and simple unsupervised DM model acquires paragraph level embeddings. We improve on the model so that we jointly learn multi-level embeddings that encode class-label topical information in addition to text.

We jointly learn a model that captures embedding representation for the target class labels, as well as word-, sentence- and document-level representations in the same space. From these multi-level embeddings we derive a set of features. Our approach requires no manual feature engineering, can cope with small amounts of labelled data and produces features that are more robust to domain variation and portable across domains.

At the document-level, the overall "topic" is a mixture of the sub-topics of paragraphs in that document. The topics of the paragraphs are in turn mixtures of the sentence topics, all the way down to

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers,
pages 2333–2343, Osaka, Japan, December 11-17 2016.

word-level semantics. Our multi-level embeddings model captures this intuition elegantly; for example, an article about *cars* might have the first sentence discussing *car manufacturing*, followed by another discussing *car safety*, etc. Each of these topics can be represented by sentence-level embeddings, while a document-level embedding can capture the overall topic of the article.

We show that classifying text based on such multi-level semantics achieves superior performance both against very specialised handcrafted models and using word sentence or document embeddings alone. We demonstrate the effectiveness of our methodology on two real-world sparsely-labelled tasks: classification of biomedical text by (i) semantic categories, and (ii) rhetorical structure.

We apply our approach at two different levels of granularity: at document-level and at sentence-level. At the sentence-level, labelled data and contexts are even more sparse. In both cases, we compare our approach under a supervised setting against a handcrafted method and show that it rivals and in some cases clearly outperforms such methods. In addition, we compare against classifiers trained using standard embedding features and show that our approach outperforms them by a large margin. We also show that fast semi-supervised classification using our multi-level embedding features achieves promising results, even when compared against an SVM classifier using standard embeddings.

To our knowledge, this is the first work to introduce multi-level embeddings for text classification and to show their superior performance against handcrafted approaches and their robustness across domains which suffer from scarcity of labelled data.

2 Related Work

Embedded distributed representations have been used widely for document and sentence classification. For example, Huang et al. (2014) learn document-level embeddings using word-level embeddings as input. Yan et al. (2015) learn document-embeddings by combing a Deep Boltzmann Machine and a Deep Belief Network. Bhatia et al. (2015) learn embeddings for large multi-label classification in situations where the label set is extremely large. Liu et al. (2015) use latent topic models to learn a topic from each word, and then learn an embedding based on both the topic and the word. Yogatama and Smith (2014) use structured regularizers based on parse trees, topics, and hierarchical word clusters, as well as hierarchical sparse coding for regularization using stochastic proximal methods (Yogatama et al., 2015).

All of these works have been trained and evaluated on general domains such as newswire rather than on sparse domains with small labelled datasets.

There are works that target small labelled data text classification in sparse domains using techniques such as active learning (Guo et al., 2013; Figueroa et al., 2012; Nissim et al., 2015). The idea of active learning is to reduce annotation effort by iteratively selecting the most informative instances to be labelled by interactively querying an expert. Although good accuracy can be achieved, the approach relies on expert knowledge and interaction, and may still require feature engineering.

Other works tackle the sparsity of labelled data using distant supervision (Reschke et al., 2014; Vivaldi and Rodríguez, 2015). Here, a classifier is trained using data labelled automatically using approximate heuristics rather than annotators. However, due to the assumptions and bias that are inherent in such labelling heuristics, this may result in lower performance.

The work presented in this paper differs from the above as it focuses on learning embeddings for sparse domains with small labelled datasets; moreover, we focus on utilizing these embeddings specifically for text classification.

3 Approach

This section first describes the Distributed Memory model (Section 3.1), and then explains how we improved it for sparse domain text classification by introducing jointly learned multi-level representations (Section 3.2).

In Section 3.3 we describe three types of features that we extract from such representations, and in Section 3.4 we explain the fixed classification setup for our task-based evaluations.

3.1 The Distributed Memory model

The Distributed Memory model is an extension of the Continuous Bag of Words (CBoW) model of Mikolov et al. (2013). The DM model learns a representation of a paragraph that captures the semantics of a paragraph's "topic". In the model, every word is represented in a word embedding matrix, and every paragraph in a paragraph embedding matrix. Paragraph representations are averaged or concatenated to predict the next word in a context using a hierarchical softmax classifier.

DM introduces an additional component to the model that allows a representation of the paragraph (via paragraph ID), which is treated internally like any other word in the model's vocabulary. It acts as a memory that remembers what is missing from the current context. The model learns a vector representation of the paragraph that captures its overall topic semantics via stochastic gradient decent.

3.2 Joint learning of multi-level embeddings

We improve DM by learning distributed representations that capture the topical information at varying levels of granularity, that is, we learn embeddings at a word-, sentence- (or paragraph-), and document-level. We also learn a distributed representation of the class labels, since these can be viewed as another level of abstraction that is more abstract than the document-level.

Our intuition is that jointly learning representations at different levels of granularity (including that of class label) provides us with better embeddings for text classification than learning a representation at each level separately. Each level captures different topic semantics, ranging from word-level to the class label. Figure 1 illustrates our model.

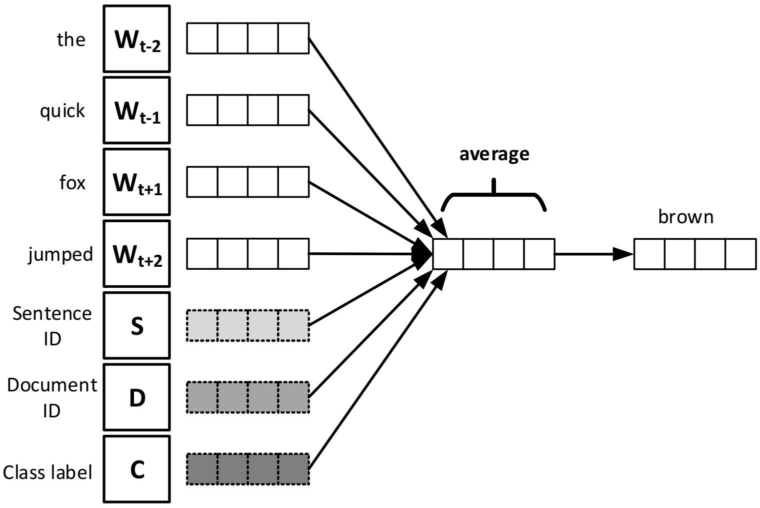

Figure 1: Illustration of distributed joint learning of different granularities of text contexts: words (W), sentences (S), documents (D) and classes (C). The model predicts the target word (w_t) based on the semantics captured by all these contexts. Shades represent level of abstraction/granularity.

In Figure 1, words from word embedding matrix **W**, sentences from sentence embedding matrix **S**, documents from document embedding matrix **D** and class-labels from class embedding matrix **C** are used as the context from which to predict the target word. That is, given a sequence of training words $w_1, w_2, w_3, ..., w_T$ that belongs to sentence s_t in document d_t, which has also a set of classification labels associated $c_1, ...c_m$. The objective of the model is to maximise the average log probability:

$$\frac{1}{T}\sum_{t=k}^{T-k} \log\ p(w_t|w_{t-k}, ...w_{t+k}, s_t, d_t, c_1, ...c_m) \tag{1}$$

We use a softmax output layer to obtain the probability of the target word given its context:

$$p(w_t|w_{t-k}, ...w_{t+k}, s_t, d_t, c_1, ...c_m) = \frac{e^{\vec{y}_{w_t}}}{\sum_i e^{\vec{y}_i}} \tag{2}$$

where each y_{w_t} is calculated as:

$$\vec{y}_{w_t} = \mathbf{U}\frac{\sum_{i=-k}^{k} \vec{w}_{t+i} + \vec{s}_t + \vec{d}_t + \sum_{i=1}^{m} \vec{c}_i}{2k+m+2} + b \tag{3}$$

where $k \neq 0$, **U** is the weight matrix, b is the bias, and we average the word vectors extracted from **W**, the sentence vectors extracted from **S**, similarly, the document vectors from **D** and class label vectors from **C**.

3.3 Extracting features

We extract three types of features from the jointly-learned multi-level representations: the sentence or document embeddings (EMBED), the distances between word embeddings (WORD-DIST) and the similarities between classes (CLASS-SIM).

Embedding features: since embeddings are learned at different levels, when classifying at the document-level, we use the document-level embeddings. Likewise for sentence-level classification, we use only the sentence-level embeddings. Word-level embeddings are only used as part of extracting distance features.

Word distance features: We measure the cosine similarity between each unique non-stop word embedding occurring in the input sentence or document with the embedding representation for a given class label, i.e., $\delta_{w_i}^{c_i} = \cos(\vec{w}_i, \vec{c}_i)$, where $\vec{w}_i$ is embedding for word w_i in the input text, and $\vec{c}_i$ is the embedding representation of a class label that has been jointly learned from the training data. Since the input text has variable length, we represent these distances in sparse vector format using a dictionary of all non-stop words in the corpus labelled with the given class c_i; i.e., a "bag of word distances".

Class-similarity features: Word distance measures capture the similarity between words and class labels, but not between phrases or sentences. For this, we use word-level embeddings to measure the semantic similarity between a class and target text (sentence or document) using the so-called Earth Mover's Distance (EMD)[1], or the energy distance of moving a distribution.

EMD has been used successfully in image retrieval (Rubner et al., 2000), document topic similarity (Wan, 2007) and more recently in combination with word embeddings (Kusner et al., 2015). This method is useful for estimating the similarity between text with varying word count and overlap: the sentence *"sipping a cup of tea"*, for example, should have a relatively small EMD compared to *"wine tasting"*, despite them having no overlap and being of different length. Kusner et al. (2015) formulate the EMD problem as a linear program that can be expressed as the following optimisation:

$$\text{emd}(d, d') = \min_{\mathbf{T} \geq 0} \sum_{i,j=1}^{n} \mathbf{T}_{ij} ||\vec{x}_i - \vec{x}_j||_2 \tag{4}$$

subject to the following flow constraints: $\sum_{j=1}^{n} \mathbf{T}_{ij} = \vec{d}_i$ and $\sum_{i=1}^{n} \mathbf{T}_{ij} = \vec{d'}_j$. Here, $\mathbf{T} \in \mathbb{R}^{n \times n}$ is a flow matrix, i.e., $\mathbf{T}_{ij}$ denotes how much of word i in the source document d travels to word j in the destination document d', and $\vec{x}_i$, $\vec{x}_j$ are embeddings for words i and j. Class-similarity features are obtained by finding the minimal distance between a given input (either document or sentence) and the given class, where only the most discriminatory word embeddings for the given class are combined, i.e., non-discriminatory words that occur in all classes are excluded[2]. We then use Equation 4 to measure the similarity between words occurring in the text and the class combined word list.

3.4 Supervised classification

We apply a fixed classification setup in order to compare our new method against several embedding baselines as well as handcrafted classification. We use Support Vector Machines with a linear kernel; implemented using scikit-learn (Pedregosa et al., 2011), and perform a standard grid search for kernel regularization parameter selection.

[1] Also known as the Wasserstein metric.

[2] We discarded all words that occur in more than 80% of all class contexts as non-discriminatory in the training set.

We use L1 and L2 normalization of input features, weighted equally according to the three aforementioned types (embedding, class-similarity and word-distance), i.e., features within each type are normalised separately and then combined.

We perform a 4-fold cross-validation setup and 5-fold nested cross-validation for kernel parameter tuning (using grid search); i.e., we do a 5-fold cross-validation grid search nested in each of the outer four folds.

3.4.1 Semi-supervised classification

In a semi-supervised setting, we use vast amounts of unlabelled data, i.e., documents/sentences unlabelled with any class information, and a much smaller amount of labelled documents/sentences. Instead of using a supervised classifier to learn the decision boundaries, we use the distance measurements and a tuned cut-off threshold for each class to determine class assignment.

We use WORD-DIST and CLASS-SIM (described in Section 3.3), and EMB-DIST: the cosine distance between the embedding of a sentence or document and an embedding of a class label. A cut-off threshold is used to determine positive or negative classification for each class. Under the WORD-DIST setup, we average all of the word distances. We perform a grid search for this threshold on 10% held-out data.

4 Task 1: Semantic text classification

We apply our methodology to a real-life biomedical text classification task. The aim of this task is to classify text at both document- and sentence-levels according to the Hallmarks of Cancer (HoC), a widely-employed framework in cancer research that was first introduced by Hanahan and Weinberg (2000). Motivated by the fact that cancer involves both genetic and epigenetic alterations (Marusyk et al., 2012), this framework provides an organizing principle to simplify the complexity of cancer biological processes (Baker et al., 2016).

4.1 Data

Baker et al. (2016) acquired a collection of PubMed abstracts using a set of search terms representative for each of the 10 hallmarks. The terms and their synonyms appearing in Hanahan and Weinberg (2000) and Hanahan and Weinberg (2011) were employed along with additional ones selected by a team of cancer researchers. Annotation was conducted by experts in cancer research, using the annotation tool described in Guo et al. (2012). Annotations are assigned at a sentence-level: a sentence is annotated if contains clear evidence relating to one or several hallmarks (Baker et al., 2016). Table 4.1 shows the distribution of 1,580 abstracts and sentences for each of the hallmark categories. The inter-annotator agreement is $k = 0.81$.

Hallmark	PS	GS	CD	RI	A	IM	GI	PI	CE	ID
# Abstracts	462	242	430	115	143	291	333	240	105	108
# Sentences	993	468	883	295	357	667	771	520	213	226

Table 1: Distribution of data for the ten hallmarks.

4.2 Handcrafted supervised model

We employ a fully supervised handcrafted baseline for this task, classifying using binary classifiers for each hallmark category. Sentences are first tokenised and part-of-speech tagged using the C&C tagger (Clark, 2002) trained on biomedical texts. The text is lemmatised using BioLemmatizer (Liu et al., 2012) and grammatical relations are extracted using the C&C Parser. The parser was trained using molecular biology annotations (Rimell and Clark, 2009). Finally, named entities are extracted from parsed data using ABNER (Settles, 2005), trained on the NLPBA and BioCreative corpora (Leitner et al., 2010).

We experimented with several types of handcrafted features for hallmark classification, chosen based on their inclusion in other state-of-the-art biomedical text classification systems. Only the first five are used for sentence-level classification, since the last two are only available at the document-level:

Lemmatised Bag of Words: the simplest feature employs all words occurring in input texts. We lemmatise the words in order to reduce sparsity.

Noun bigrams: Noun bigrams are used because they can be useful in capturing two word-concepts in texts (e.g., *Gene silencing*).

Grammatical relations: we use the *dobj* (direct object), *ncsubj* (non-clausal subject), and *iobj* (indirect object) relations, plus the head and dependent words in relations.

Verb classes: verb classes group semantically similar verbs together, abstracting away from individual words when faced with data sparsity. We used the hierarchical classification of 399 verbs by Sun and Korhonen (2009).

Named entities: domain-specific concepts, providing another way to group bags of words into meaningful categories. We use five types which are particularly relevant for cancer research: Proteins, DNA, RNA, Cell Line, and Cell Type.

Medical Subject Headings (MeSH): is a comprehensive controlled vocabulary for indexing journal articles and books in the life sciences. Most abstracts in our dataset contain an associated list of MeSH terms which we employ as features.

Chemicals list: a total of 3,021 associated chemicals (manually annotated). We use these as features, since processes involved with hallmarks might involve similar chemicals.

5 Task 2: Rhetorical text classification

Rhetorical text classification (also known as information structure analysis) segments scientific text into information categories. One such classification technique is argumentative zoning (Teufel and Moens, 2002) which captures the rhetorical progression of the scientific argument by segmenting a document into several zones, such as: “Objective”, “Background”, “Method”, “Result”, and “Conclusion”.

This task differs from Task 1 in that the objective is to classify scientific text according to generic labels (i.e., unrelated to domain-specific knowledge) and the focus is on a different classification features, such as the position of the text in the document and the author’s writing style. For example, the “Objective” zone of the argument generally appears very early in the article using an active voice.

5.1 Data

We evaluate using an expert-annotated dataset from (Guo et al., 2010) comprising of 1000 PubMed abstracts relevant to cancer biology. The dataset consists of 7985 labelled sentences, with an inter-annotator agreement of $k = 0.85$. There are five mutually non-exclusive classes, described together with their frequencies in Table 5.1.

Class	Description	# Abstracts	# Sentences
Objective (*OBJ*)	The background and the aim of the research	744	812
Background (*BKG*)	The circumstances pertaining to the current work	692	1517
Method (*METH*)	The way to achieve the goal	640	1617
Result (*RES*)	The principal findings	889	4028
Conclusion (*CON*)	Analysis, discussion and the main conclusions	859	1484

Table 2: Description of argumentative zones and their distribution in the annotated data.

5.2 Handcrafted supervised model

Many of the features used for this task are similar to those used in the Task 1, namely Bag-of-Words, Bigrams, Grammatical Relations. Here we also include Part-of-Speech tags, and the following task-specific features:

Location: categories tend to appear in typical positions in a document, e.g., *BKG* usually occurs at the beginning and *CON* at the end. The abstract is divided into ten equal parts and the location of a sentence is defined by the parts where the sentence begins and ends.

History: the category of the preceding sentence is used as a feature. This is because certain categories tend to appear before others. For example, *RES* tends to be followed by *CON* rather than other categories.

Voice: there is a correlation in scientific writing between the active and passive voice and certain categories, for example, passive voice is more frequent in *METH*.

6 Results

We now present the results of our experiments, where we compare our method to the handcrafted models, in addition to several baselines detailed below.

We train Skip-Gram with Negative Sampling (SGNS) representations on the corpus, and obtain sentence or document-level embedding using a composition function $f(w_i, ...w_n)$, where f is either addition (ADD), averaging (AVG) or the maximum (MAX). We do the same with Continuous Bag of Words (CBoW) representations. The resultant composed embeddings are used as input features for the classifier. For conciseness, we include only the best performing composite function here.

We also implemented using Keras (Chollet, 2015) a Convolutional Neural Network (ConvNet) for both sentence and document classification. We trained a binary classifier for each class, each consisting of the following layers: (i) input layer (domain trained embeddings using SGNS with $dim = 200$), (ii) 1-dimensional convolutional layer, (iii) max pooling layer with $dropout = 0.5$, (v) fully connected layer, and (vi) a binary softmax output layer. We use a binary cross-entropy loss function, and the Adam optimizer (Kingma and Ba, 2014). We also experimented with two key ConvNet parameters: the number of filters and the filter window size.

Finally, we compare against standard Bag of Words (BoW) classification, where each non-stop word in the corpus is a binary feature.

6.1 Task 1 results

The aim of Task 1 (semantic text classification) is to classify text into ten mutually non-exclusive classes, the Hallmarks of Cancer. The task has two sub-tasks: document-level and sentence-level classification. Table 6.1 shows the results for both levels. Sentence-level classification is more difficult, due to the smaller context information available. The table shows the results for the composed embedding baselines, the supervised BoW baseline and the handcrafted supervised model, as described in Section 4.2. This is followed by the three feature types (EMBED, CLASS-SIM, WORD-DIST) in all possible combinations and finally the full model, i.e., the one that uses all features.

With regards to the EMBED feature type, we distinguish between learning the representation independently (e.g., embeddings are learned without knowledge of the document) or jointly as described in Figure 1. We can see that the EMBED features by themselves perform better than any of the embedding baseline models. Jointly learning embeddings improves the F-score by approximately 4-5% for both document and sentence classification.

When considering the three features types, CLASS-SIM outperforms both EMBED and WORD-DIST, with an especially notable improvement in document classification.

When pairing the three feature types, the combination CLASS-SIM + WORD-DIST gives the best results, as is consistent with their individual results. Finally, when combining all three features, the full model outperforms all baselines with a significant margin, especially notable for sentence-level classification. Regarding the semi-supervised models, using class similarity CLASS-SIM alone significantly outperforms using word cosine distance WORD-DIST, and document-embedding to class-embedding distance EMB-DIST.

6.2 Domain variation

We also investigate the performance of our model and baselines when subjected to domain variation; that is, when we learn the embeddings from a different domain than that of the classification task. We experimented by learning the embeddings using the Wikipedia corpus. We seed the model with the labelled HoC training data and, then train on Wikipedia instead of domain specific literature acquired from PubMed.

	Document classification			**Sentence classification**		
Model	Precision	Recall	F-score	Precision	Recall	F-score
SGNS	43.7	25.9	32.5	13.6	23.3	17.2
CBoW	30.9	27.6	29.2	7.5	15.1	15.0
BoW	47.9	37.9	42.3	53.8	24.1	33.3
ConvNet	80.9	60.7	69.4	55.4	43.8	48.9
Handcrafted	82.8	69.4	75.5	59.2	46.4	51.4
EMB-DIST (semi-supervised)	24.3	28.5	26.3	25.4	25.4	21.9
WORD-DIST (semi-supervised)	30.9	36.5	33.5	43.7	24.6	31.5
CLASS-SIM (semi-supervised)	**44.1**	**38.8**	**41.3**	**26.7**	**42.1**	**32.6**
EMBED (independently)	44.0	37.4	40.4	26.5	48.0	34.2
EMBED (joint training)	54.2	46.4	49.9	37.6	39.6	38.6
CLASS-SIM	80.2	49.9	59.4	36.2	45.8	40.5
WORD-DIST	58.5	51.9	55.0	32.7	40.3	36.1
EMBED + CLASS-SIM	69.3	58.3	63.3	54.7	52.1	53.3
EMBED + WORD-DIST	60.9	60.4	60.7	54.6	56.3	55.4
CLASS-SIM + WORD-DIST	64.5	72.7	68.4	61.5	61.0	61.2
EMBED + CLASS-SIM + WORD-DIST	**85.5**	**69.8**	**76.4**	**77.7**	**60.1**	**67.6**

Table 3: Task 1 performance comparison. All figures are micro-averages (%).

Naturally, we expect all of the models to perform worse with Wikipedia-trained embeddings than with domain specific embeddings. This is indeed what happens (Table 6.2); however, some models prove more robust than others, i.e., their drop in F-score accuracy is smaller. By this measure, our full model and the semi-supervised models are less susceptible to domain variation with both document and sentence-level classification.

	Document		Sentence	
Model	Domain	Wikipedia	Domain	Wikipedia
SGNS	32.5	18.4	17.2	11.1
CBoW	29.2	14.3	15.0	10.4
ConvNet	69.4	38.3	48.9	29.7
Semi-supervised [3]	41.3	35.3	32.6	28.6
Full model	**76.4**	**61.5**	**67.6**	**54.6**

Table 4: Document and sentence classification micro-averaged F-score (%) using domain-specific and Wikipedia embeddings.

6.3 Task 2 results

The objective of Task 2 is to classify scientific text according to five argumentative zones. Table 6.3 summarises the results. Similar to Task 1, all three feature types perform significantly better than the embedding baseline models. When analysing the three feature types separately, WORD-DIST outperforms the other two. EMBED + WORD-DIST is the best-performing feature pair.

The full model significantly outperforms all baselines. However, it does not match the handcrafted approach. This is because the most influential feature in this task is the location of the text (Guo et al., 2011; Kiela et al., 2015). As our model does not take any word or sentence ordering into account, it would be difficult to compensate for the location feature. If, however, we include the location feature in addition to the three feature types in the SVM classification, our model outperforms the handcrafted

[3] Semi-supervised model uses CLASS-SIM.

baseline by a 1.4% difference. Admittedly, this would make the model slightly handcrafted by itself, but no additional work is necessary to get this feature and it does not vary across tasks or domains. This shows that our model including location information provides better features for this task than the handcrafted approach including location information. Looking at the semi-supervised models, the results suggest that CLASS-SIM outperforms the other feature types by an even larger margin than for Task 1.

Model	**Precision**	**Recall**	**F-score**
SGNS	45.6	31.5	37.3
CBoW	47.3	30.9	37.4
BoW	54.8	35.1	42.7
ConvNet	74.9	66.9	70.7
Handcrafted	88.9	85.0	86.9
EMB-DIST (semi-supervised)	23.7	38.9	29.4
WORD-DIST (semi-supervised)	36.6	28.8	32.2
CLASS-SIM (semi-supervised)	**57.1**	**40.9**	**47.7**
EMBED (sentences only)	43.3	41.9	42.6
EMBED (joint training)	57.6	37.7	45.6
CLASS-SIM	58.7	46.0	51.6
WORD-DIST	55.2	51.8	53.5
EMBED + CLASS-SIM	64.5	59.9	62.1
EMBED + WORD-DIST	78.1	57.5	66.3
CLASS-SIM + WORD-DIST	82.0	54.5	65.5
EMBED + CLASS-SIM + WORD-DIST	**81.2**	**72.7**	**76.7**
Full model + location	**89.6**	**86.9**	**88.3**

Table 5: Task 2 Micro-averaged performance comparison. All figures are percentages.

7 Discussion and conclusions

The aim of this paper has been to produce a robust approach to text classification for domains suffering from sparsity of labelled data, and to alleviate the necessity for handcrafting features. Our novel methodology jointly learns distributed semantic representations at the level of words, sentences, documents and class.

The intuition is that embeddings at each level capture slightly different topical semantics. We therefore employ these embeddings to produce three types of features that require no additional data or labour, that are efficient to extract and much easier to port than handcrafted features. We have shown how these feature types can be used with standard classification algorithms such as SVMs and with semi-supervised classification where the decision boundaries are not learned from labelled data.

In the first task (semantic text classification) our approach matched or outperformed a handcrafted fully-supervised approach. The model performed substantially better at sentence-level classification which had much less context than the document-level classification. We also showed that our features are less susceptible to domain variation.

In the second task (rhetorical text classification), the proposed model outperformed all baselines, as well as the handcrafted approach when including location information in the classification process.

Acknowledgments

The first author is funded by the Commonwealth Scholarship and the Cambridge Trust. This work is supported by Medical Research Council grant MR/M013049/1 and the Google Faculty Award.

References

Charu C Aggarwal and ChengXiang Zhai. 2012. A survey of text clustering algorithms. In *Mining Text Data*, pages 77–128. Springer.

Simon Baker, Ilona Silins, Yufan Guo, Imran Ali, Johan Högberg, Ulla Stenius, and Anna Korhonen. 2016. Automatic semantic classification of scientific literature according to the hallmarks of cancer. *Bioinformatics*, 32(3):432–440.

Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. 2015. Sparse local embeddings for extreme multi-label classification. In *Advances in Neural Information Processing Systems*, pages 730–738.

François Chollet. 2015. Keras: Deep learning library for theano and tensorflow.

Stephen Clark. 2002. Supertagging for combinatory categorial grammar. In *Proceedings of the 6th International Workshop on Tree Adjoining Grammars and Related Frameworks (TAG+ 6)*, pages 19–24.

Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. 2007. Transferring naive bayes classifiers for text classification. In *Proceedings of the national conference on artificial intelligence*, volume 22, page 540. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Rosa L Figueroa, Qing Zeng-Treitler, Long H Ngo, Sergey Goryachev, and Eduardo P Wiechmann. 2012. Active learning for clinical text classification: is it better than random sampling? *Journal of the American Medical Informatics Association*, 19(5):809–816.

Yufan Guo, Anna Korhonen, Maria Liakata, Ilona Silins Karolinska, Lin Sun, and Ulla Stenius. 2010. Identifying the information structure of scientific abstracts: an investigation of three different schemes. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 99–107. Association for Computational Linguistics.

Yufan Guo, Anna Korhonen, Ilona Silins, and Ulla Stenius. 2011. Weakly supervised learning of information structure of scientific abstractsis it accurate enough to benefit real-world tasks in biomedicine? *Bioinformatics*, 27(22):3179–3185.

Yufan Guo, Ilona Silins, Roi Reichart, and Anna Korhonen. 2012. CRAB reader: A tool for analysis and visualization of argumentative zones in scientific literature. In *Proceedings of COLING 2012: Demonstration Papers*, pages 183–190.

Yufan Guo, Ilona Silins, Ulla Stenius, and Anna Korhonen. 2013. Active learning-based information structure analysis of full scientific articles and two applications for biomedical literature review. *Bioinformatics*, 29(11):1440–1447.

Douglas Hanahan and Robert A Weinberg. 2000. The hallmarks of cancer. *Cell*, 100(1):57–70.

Douglas Hanahan and Robert A Weinberg. 2011. Hallmarks of cancer: the next generation. *Cell*, 144(5):646–674.

Chaochao Huang, Xipeng Qiu, and Xuanjing Huang. 2014. Text classification with document embeddings. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 131–140. Springer.

Douwe Kiela, Yufan Guo, Ulla Stenius, and Anna Korhonen. 2015. Unsupervised discovery of information structure in biomedical documents. *Bioinformatics*, 31(7):1084–1092.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Q. Weinberger. 2015. From word embeddings to document distances. In David Blei and Francis Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 957–966. JMLR Workshop and Conference Proceedings.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196.

Florian Leitner, Scott A Mardis, Martin Krallinger, Gianni Cesareni, Lynette A Hirschman, and Alfonso Valencia. 2010. An overview of biocreative ii. 5. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 7(3):385–399.

Haibin Liu, Tom Christiansen, William A Baumgartner Jr, and Karin Verspoor. 2012. Biolemmatizer: a lemmatization tool for morphological processing of biomedical text. *J. Biomedical Semantics*, 3:3.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical word embeddings. In *AAAI*, pages 2418–2424.

A. Marusyk, V. Almendro, and K. Polyak. 2012. Intra-tumour heterogeneity: a looking glass for cancer? *Nature Reviews Cancer*, 12(5):323–334.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Nir Nissim, Mary Regina Boland, Robert Moskovitch, Nicholas P Tatonetti, Yuval Elovici, Yuval Shahar, and George Hripcsak. 2015. An active learning framework for efficient condition severity classification. In *Artificial Intelligence in Medicine*, pages 13–24. Springer.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.

Changqin Quan, Meng Wang, and Fuji Ren. 2014. An unsupervised text mining method for relation extraction from biomedical literature.

Kevin Reschke, Martin Jankowiak, Mihai Surdeanu, Christopher D Manning, and Daniel Jurafsky. 2014. Event extraction using distant supervision. In *Language Resources and Evaluation Conference (LREC)*.

Laura Rimell and Stephen Clark. 2009. Porting a lexicalized-grammar parser to the biomedical domain. *Journal of biomedical informatics*, 42(5):852–865.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121.

Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.

Burr Settles. 2005. Abner: an open source tool for automatically tagging genes, proteins and other entity names in text. *Bioinformatics*, 21(14):3191–3192.

Matthew S Simpson and Dina Demner-Fushman. 2012. Biomedical text mining: a survey of recent progress. In *Mining text data*, pages 465–517. Springer.

Lin Sun and Anna Korhonen. 2009. Improving verb clustering with automatically acquired selectional preferences. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 638–647. Association for Computational Linguistics.

Simone Teufel and Marc Moens. 2002. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational linguistics*, 28(4):409–445.

Jorge Vivaldi and Horacio Rodríguez. 2015. Medical entities tagging using distant learning. In *Computational Linguistics and Intelligent Text Processing*, pages 631–642. Springer.

Xiaojun Wan. 2007. A novel document similarity measure based on earth movers distance. *Information Sciences*, 177(18):3718–3730.

Yan Yan, Xu-Cheng Yin, Sujian Li, Mingyuan Yang, and Hong-Wei Hao. 2015. Learning document semantic representation with hybrid deep belief network. *Computational intelligence and neuroscience*, 2015.

Dani Yogatama and Noah A Smith. 2014. Linguistic structured sparsity in text categorization.

Dani Yogatama, Manaal Faruqui, Chris Dyer, and Noah Smith. 2015. Learning word representations with hierarchical sparse coding. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 87–96.

Shaodian Zhang and Noémie Elhadad. 2013. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *Journal of biomedical informatics*, 46(6):1088–1098.