EClaiRE: Context Matters! – Comparing Word Embeddings for Relation Classification

Lena Hettinger[1] Albin Zehe[1] Alexander Dallmann[1] Andreas Hotho[1]

Abstract: In recent years, there has been an increasing interest in the task of relation classification, which aims to label a relation between two semantic entities. In this work, we investigate how domain-specific information influences the performance of ClaiRE, an SVM-based system combining manually crafted features with word embeddings. To this end, we experiment with a wide range of word embeddings and evaluate on one general and two scientific relation classification datasets. We release all of our code for relation classification and data for scientific word embeddings to enable the reproduction of our experiments.[2]

Keywords: word embedding; relation classification; context sensitive; domain specific

1 Introduction

Finding an appropriate representation for a word is a challenging task in Natural Language Processing, especially considering the fact that words can have multiple meanings. Word ambiguity becomes most apparent when looking at datasets from different domains. For example, the word *string* can either denote a "string of cotton" or a "string of characters", depending on whether it appears in its most common or a domain specific context like computer science. In this paper, we want to examine the role of ambiguous or domain-specific expressions for the relation classification task (cf. Sect. 4).

The goal of relation classification is to label the semantic relation between two selected entities in a sentence. There are two reasons why this is a fitting task to examine word representations. First, there exist indications that a semantically correct representation is important for good performance [He18]. Second, it is obvious that a certain relation described between two entities might not correspond to the most general meaning of a word, as exemplified in Fig. 1. Identifying words only with their most common meaning, the sentence in the example would not make sense and the expressed relation would be unclear.

Since our focus is more on understanding the influence of domain-specific word senses than on providing a new state of the art, we decided to use an SVM-based model rather than a neural network, which requires far less computational power and therefore enables us to

[1] University of Wuerzburg, DMIR Group, Am Hubland, 97074 Würzburg, Germany {hettinger,zehe,dallmann, hotho}@informatik.uni-wuerzburg.de

[2] https://gitlab2.informatik.uni-wuerzburg.de/dmir/eclaire

doi:10.18420/inf2019_24

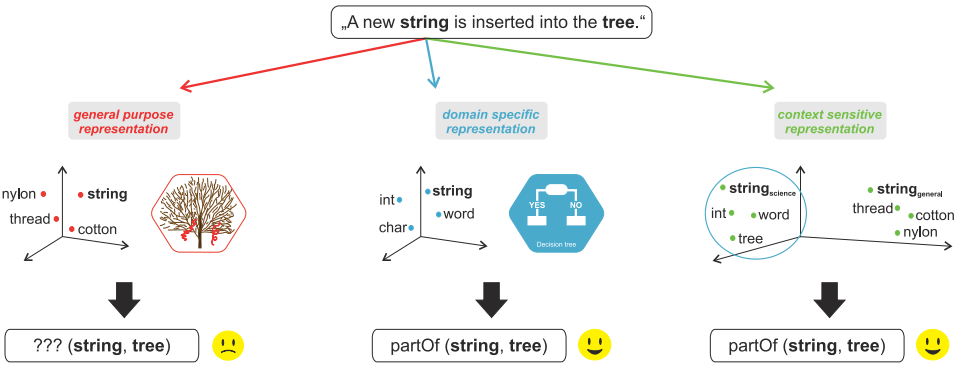


Fig. 1: Problems in relation classification arising from word representations that do not reflect domain-specific meaning.

investigate a wider range of settings. Specifically, we use ClaiRE [He18], an SVM based on both hand-crafted features and word embeddings. ClaiRE has already been shown to be highly dependent on the word embeddings used to encode the input, making it a suitable model for our research.

We investigate a wide range of word embeddings, which can generally be partitioned into three classes: (a) Publicly available static embeddings trained on general corpora, (b) our own domain specific, static embedding trained on a corpus of scientific articles and (c) publicly available context-sensitive embeddings trained on a general corpus.

We show that domain-specific word embeddings outperform general ones on the scientific domain, and, conversely, that specialised word embeddings can not be transferred to the general domain. We also find that context-sensitive embeddings outperform both types of static embeddings on each domain, but further improvements can be reached by combining them with domain-specific embeddings.

2 Related Work

Relation classification is an interesting topic of research, as relations between pairs of entities are studied across a wide range of domains. It was the topic of several challenges, e. g. SemEval-2010 (general domain) [He09] and SemEval-2018 (scientific domain) [Gá18]. Although neural network based approaches currently claim state of the art on both domains [RHZ18, Wa16], SVM based approaches with lexical and semantic features have shown competitive performance in the past [He18, RH10]. For a more in-depth coverage of related work on relation classification we refer to [He18, RHZ18]. In this work we investigate the dependency of relation classification on feature and task domain. Hence, we do not focus on achieving state-of-the-art results.

Word embeddings have been proven effective for a wide range of NLP tasks [Ki14, Ma14]. Consequently, much work has been done on word embeddings in recent years. We will provide an overview of some commonly used models in Sect. 3.

Although word embeddings often improve the performance of downstream tasks, word embeddings derived from general corpora can be suboptimal if used in specialised domains [Ta14]. For example, word embeddings trained on a domain-specific corpus improve relation classification performance in the scientific domain [He18]. However, large corpora needed for training a word embedding model from scratch are not always available. As a result, some work has been done on leveraging multiple corpora by training cross-domain embeddings [BMiK15, YLZ17]. Another direction of research focuses on adapting pre-trained language models to a specific domain [HR18, LL13, Ni17, Yu17].

Commonly used word embeddings provide a single static vector for every word in the vocabulary. As a result, different word senses are not accurately represented. This issue has been addressed by modifying existing models to represent a word by multiple sense vectors [IPN15, CLS14, JP15]. However, these models still suffer from limitations, for example by relying on a limited semantic network [JP15]. A recent line of research addresses these limitations by learning context-sensitive embeddings that compute a word representation dependent on the context, e.g. the sentence [CCP18]. Evaluation of context-sensitive embeddings is mostly focused on the general domain [ABV18, Mc17, Pe18]. In contrast we aim to investigate the suitability of general context-sensitive embeddings for a domain specific task and compare the performance to domain-specific static embeddings.

3 Background: Word Embeddings

In order to analyse the connection between the domain of relation classification and word embeddings, we first need to take a look at the data and models used to create them. This will later enable us to pose some hypotheses as to why certain embeddings work better for one or the other context. This chapter provides an overview over the different word embeddings we use in this work. We classify them into three groups: Traditional embeddings, domain-specific embeddings and context-sensitive embeddings, as shown in Tab. 1.

3.1 Traditional Embeddings

The first group of embeddings are publicly available sets of vectors that are commonly used in NLP. There exist multiple algorithms for the creation of these embeddings, where some of the most commonly used are word2vec [Mi13], GloVe [PSM14] and FastText [Bo17]. We use publicly available vectors trained with each of these algorithms, as well as ConceptNet Numberbatch [SCH17], which is a combination of the three aforementioned embeddings and the knowledge graph ConceptNet. The first part of Tab. 1 provides some details about the

traditional embeddings used in this work. We retrieve all of these through `gensim-data`[3], a data repository for pre-trained NLP models.

WE	dim	Origin	size
w2v	300	Google	100.0
GloVe	300	WP, Gigaword	6.0
FastText	300	WP, Gigaword, ...	16.0
CNB	300	w2v, GloVe, ...	n.a.
$w2v_{arXiv}$	300	arXiv	0.7
ELMo	3072	WP, WMT	5.5
Flair	4096	WMT	0.8

Tab. 1: Details about the pretrained word embeddings used in this paper. WP: Wikipedia, WMT: Workshop on Statistical Machine Translation. Size of data sets is given in billion tokens.

3.2 Domain-specific Embeddings

To generate domain-specific $w2v_{arXiv}$ embeddings, we use word2vec on a large corpus of scientific papers. We downloaded LaTeX sources for all papers published in 2016 on arXiv.org using the provided dumps.[4] We converted LaTeX sources to plain text using a manually crafted set of regular expressions. We refer to our source code for details about the conversion and to [He18] for further details about constructing scientific embeddings.

3.3 Context-Sensitive Embeddings

Context-sensitive embeddings have been on the rise since the publication of CoVe [Mc17]. They represent a change of paradigm, as words are not described by static vectors but are assigned a different vector depending on the sentence they appear in. The advantage of this approach is that ambiguous words do not have to be resolved separately. Instead the model is able to distinguish between the different meanings using the word's context. In this work we compare traditional embeddings with the two currently best-performing context-sensitive models: ELMo, which has shown to perform better than CoVe, and Flair, which outperforms ELMo for some tasks in conjunction with character features and/or traditional embeddings. These embeddings make up the third part of Tab. 1.

[3] `https://github.com/RaRe-Technologies/gensim-data`
[4] `https://arxiv.org/help/bulk_data`

3.3.1 ELMo

ELMo (Embeddings from Language Models) [Pe18] is a deep bidirectional language model that produces character based word vectors from its internal states. More specifically, ELMo computes multiple representations with different amounts of context and semantics in its layers.

First, the model builds context insensitive word representations ($ELMo_0$) by applying a character based CNN on every word in the sentence. The context sensitive embeddings ($ELMo_1$, and $ELMo_2$ respectively) stem from a 2-layer biLSTM [HS97, SP97] that takes the previously computed word representations ($ELMo_0$) as input. Ultimately, the biLM provides three layers of representations for any input token, forming a hypercolumn of the three vectors ELMo = [$ELMo_0$, $ELMo_1$, $ELMo_2$].

ELMo claims that lower-level LSTM states capture aspects of syntax, while higher-level states model aspects of word meaning in context. It has proven its success on different tasks such as question answering, semantic role labeling and named entity extraction. However, it has not been applied to the task of relation classification before. To the best of our knowledge, the performance of ELMo across different domains of data has not been researched yet explicitly. For our experiments, we use an officially published ELMo model[5] that has been pre-trained on a 5.5 billion token general dataset and produces context sensitive token representations ELMo $\in \mathbb{R}^{3072}$.

3.3.2 Flair

Flair [ABV18] is a context-sensitive model which claims to outperform ELMo on tasks such as named entity recognition and chunking. Similar to ELMo it leverages the internal states of a trained language model, but uses characters as atomic units for a 1-layer biLSTM. Flair is thus trained without any explicit notion of words and at each point in the sequence predicts the next character. This is different from the character-aware LM used by ELMo, which operates on word level and character convolutions.

In this work we utilise Flair embeddings trained on the 1-billion word corpus [Ch13].[6] A Flair vector consists of the 2048 hidden states of a forward and backward LSTM, which can be described as a hypercolumn: Flair = [$Flair_f$, $Flair_b$] $\in \mathbb{R}^{4096}$. We try different combinations of Flair vectors (as well as for ELMo), to see which best fits the relation classification task in Sect. 6.3.

[5] `https://allennlp.org/elmo`, (Original 5.5B)

[6] `https://github.com/zalandoresearch/flair/blob/master/resources/docs/TUTORIAL_WORD_EMBEDDING.md`, (news-forward/ -backward)

4 Task: Relation Classification

We will now describe the task of relation classification and the model we utilise to investigate the effect of task and feature domain in detail.

4.1 Task Description

The goal of relation classification is to classify semantic relations between entities into a predefined set of categories. In order to further illustrate the problem we are dealing with in this paper, we picture two specific relation samples for each domain, general and scientific text, in Tab. 2. Relations are marked as reversed, if the order their entities appear in does not match the class order (cf. Sect. 4.2).

Across domains, some tokens have ambiguous meaning. For example, while the word "paper" is commonly associated with a material, in the scientific domain it will mostly stand for a publication. As words may be linked to specific relation classes, representing different appearances with a single static vector and thus ignoring the context they appear in might lead to misclassification.

Domain	Label	Sample
General	Component-Whole	The tailpiece anchors the strings to the lower **bout** of the **violin** by means of the tailgut.
General	Instrument-Agency (rev)	Stanford **researchers** have coated **paper** with carbon nanotubes [. . .].
Science	Result	**Combination methods** are an effective way of improving **system performance**.
Science	Usage (rev)	In this paper we describe a speaker dependent system for predicting segmental **duration** from **text** [. . .].

Tab. 2: Examples for relation classification samples from the general and scientific domain. Relation entities are denoted by bold font.

4.2 Feature Extraction: ClaiRE

We will use ClaiRE[7] as a base system for relation classification. ClaiRE is based on an SVM trained on a combination of word embeddings with manually crafted features. We selected this method as the original paper has shown that both the manual features and the word embeddings are critical for the performance on SemEval-2010 Task 8. Thus, it is reasonable to keep the manual features fixed while swapping different word embeddings to compare their performance.

[7] `https://gitlab2.informatik.uni-wuerzburg.de/dmir/claire`

In Tab. 3, we provide a short overview on hand-crafted lexical features constructed from text and numeric features based on word embeddings, for more details see [He18]. When constructing features for relation classification, the relevant parts of a sentence consist of the two entities that are part of a relation and their context, meaning the words in between entities. We distinguish between a *start* and *end entity* of a relation. If the start entity appears after the end entity within a sentence, the direction of a relation is **reversed**, as can be seen in Tab. 2.

Some features are slightly modified in this work, noted by a star in Tab. 3. In contrast to [He18], we did not utilise SpaCy[8] for preprocessing text and treated *dist* and *sim* as numeric features instead of boolean to provide more information. We utilise the WordNet Lemmatizer of nltk[9] for lemmatisation and the Stanford POS Tagger [To03] for Part-of-Speech (POS) tags. Before computing features, all lemmatised context words below a certain threshold were discarded to limit the vocabulary. Preliminary tests have shown that the optimal lemma frequency is 5 for both datasets.

Feature Set	Description
*bow**	BOW (lemmatised) from context
*pos**	Stanford POS tags from context
*pospath**	concatenated POS tags from context
*dist**	number of words in context
lc	Levin classes of verbs in context
ents	entity (head) without order
startEnt	entity (head)of relation start
endEnt	entity (head) of relation end
c	embedding vector of context
e_1	embedding vector of first entity
e_2	embedding vector of second entity
*sim**	similarity score of two entity vectors
simb	similarity bucket of similarity score

Tab. 3: Generated features for use in relation classification, grouped by type: lexcial context, lexical entity and embedding features. Features which differ slightly from [He18] are marked by *.

5 Datasets: General and Scientific Relations

Since we want to compare the performance of different word embeddings across domains, we need datasets from different domains of text. There have been multiple SemEval tasks concerned with the task of relation classification in the past, some on general corpora and some on data from specialised domains. We use the dataset from SemEval-2010 Task 8 (SE10-8) as an example of a "general" corpus and the dataset from SemEval-2018 Task 7 for a specialised corpus, in particular for the scientific domain.

[8] `https://spacy.io/`
[9] nltk.stem.wordnet.WordNetLemmatizer

SE10-8 consists of 10 717 samples of semantic relations between nominals in a sentence, collected by pattern-based web search, where 8000 samples are used for training (SE10-8_{train}) and 2717 as a test set (SE10-8_{test}). Each sample is labelled with one of 10 classes, see [He09] for a detailed description. In addition to the relation label, the direction of a relation has to be predicted for this task. We decided to model the direction as part of the label (e. g. *Cause-Effect-Reverse*) instead of using a two-stage approach (i.e., predicting the label and the direction separately) as initial experiments have shown that this performs better for ClaiRE.

The relation classification task (task 7) of SemEval-2018[10] is comprised of two subtasks. In the first subtask participants were provided with 1228 training samples and a test set (SE18-7_{clean}) with 355 relations, where both entities and relations were manually labelled. The second subtask consists of 1245 training samples and a different test set (SE18-7_{noisy}) with 355 relations, but here entities have been extracted automatically, thus introducing noise. Samples for both subtasks stem from abstracts from the ACL Anthology Corpus.

Combining both training sets has been shown to improve performance on both subtasks [He18], thus we form our final training set (SE18-7_{train}) by combining the training samples from both tasks. The final training set then has 2473 samples and each sample belongs to one of the six domain-specific classes in Tab. 4. As the relation direction is not part of the classification task it can be utilised as a feature in this case.

Note that both test sets SE18-7_{clean} and SE18-7_{noisy} contain classes (TOPIC, COMPARE) that are heavily underrepresented. This leads to some artifacts in the evaluation score for this dataset, which we will discuss in Sect. 6.2.

label	**SE18-7_{clean}**		**SE18-7_{noisy}**	
COMPARE	21	5.9 %	3	0.8 %
MODEL-F.	66	18.6 %	75	21.1 %
PART_W.	70	19.7 %	56	15.8 %
RESULT	20	5.6 %	29	8.2 %
TOPIC	3	0.8 %	69	19.4 %
USAGE	175	49.3 %	123	34.6 %

Tab. 4: Distribution of class labels for the SE18-7 datasets with absolute values and relative frequency.

6 Comparison of Embeddings and Datasets from Different Domains

We will now describe the experimental setup used in our relation classification evaluation and the results we obtained.

[10] https://competitions.codalab.org/competitions/17422

6.1 Experimental Setup

In order to assess the relative quality of different embeddings for a domain, we follow the setting in [He18], using an rbf-SVM as a base classifier and the features described therein with changes as noted in Sect. 4.2. We keep the hand-crafted features fixed while varying the embedding-based features, constructing them from our different embeddings. We also experiment with using a combination of multiple embeddings. In this case, we construct all embedding features using the concatenation of the embeddings.

To further strengthen our model, we use an ensemble of 10 SVMs with shuffled training data. As we utilise the probability estimates of an SVM to predict test labels [WLW04], we average over all probabilities in the ensemble and predict the class with the highest score. We use macro-averaged F1-score to evaluate our models, as the rating in both of the SemEval tasks was based on this score. We made use of the respective official evaluation scripts to compute the scores.

6.2 Classification Results

We report overall results for all three datasets in Tab. 5 before taking a closer look at different embeddings.

The first line reports the best result achieved on the respective datasets by any previously presented system. The best systems rely on rather complicated neural networks that are specifically tuned to the task, requiring large amounts of computational power. To enable fair comparison against an SVM-based system, the next line shows the best SVM so far as a baseline. Taking only lexical features into consideration and excluding word embeddings completely (w/o WE), ClaiRE exhibits insufficient performance, once again proving the worth of word embeddings for the task of relation classification. On the other hand, using only word embedding features and ignoring the lexical features (only WE) already performs rather well.

The next part of the table shows the performance of ClaiRE with static word embeddings (word2vec, GloVe, CNB and FastText) in combination with hand-crafted features. The embeddings have been pre-trained on large general corpora (see Sect. 3). Our results show that by utilising these embeddings, ClaiRE performs quite well on the SE10-8 dataset from the general domain, but the performance deteriorates on the scientific datasets from SE18-7.

The opposite effect can be found for the domain-specific embedding. As expected, $w2v_{arXiv}$ greatly outperforms traditional WEs on the scientific domain. On the general SE10-8 data, however, the scientific embedding performs far worse than the general embeddings.

The final part of the table shows the performance of the context-sensitive embeddings ELMo and Flair. For this summary we utilise the first context-sensitive ELMo-layer ($ELMo_1$) and

the Flair backward-layer (Flair$_b$). We will take a closer look at embedding-layers in Sect. 6.3 and show the reason for that decision. While both ELMo and Flair consistently perform well on both domains, ELMo achieves better scores on both the SE10-8 and SE18-7$_{\text{clean}}$ dataset.

Model	**SE10-8**	**SE18-7** clean	**SE18-7** noisy
best	*88.00*[a]	*81.72*[b]	*90.40*[b]
best SVM	82.19[c]	75.11[d]	81.44[d]
w/o WE	73.15	68.58	74.10
only WE	82.42[e]	76.27[e]	**85.79**[f]
w2v	78.46	72.28	77.56
GloVe	77.45	67.03	81.21
CNB	79.20	72.18	79.12
FastText	79.50	69.21	81.57
w2v$_{\text{arXiv}}$	74.90	77.76	84.47
ELMo$_1$	83.22	80.34	84.21
Flair$_b$	79.01	77.32	85.38
best 2-WE	**83.81**[g]	**81.13**[h]	85.63[g]

[a] [Wa16] [b] [RHZ18] [c] [RH10] [d] [He18]
[e] ELMo$_1$ [f] w2v$_{\text{arXiv}}$ [g] ELMo$_1$/Flair$_b$
[h] ELMo$_1$/w2v$_{\text{arXiv}}$

Tab. 5: Results from relation classification on three datasets using different embeddings. Results are given as macro-averaged F1-scores. The best result achieved by an SVM for each dataset is marked in bold. Footnotes denote the embedding used on the respective datasets.

We also evaluated combinations of lexical features and two embeddings. The results are shown in the last line of the table (best 2-WE)[11]. Again, context-sensitive ELMo embeddings contribute substantially to a very good performance, appearing in the best WE-pair of each dataset.

Overall, ELMo is the best-performing embedding for two out of three datasets. The automatically built dataset SE18-7$_{\text{noisy}}$ forms an exception, as domain-specific w2v$_{\text{arXiv}}$ vectors perform best for this task.But as mentioned in Sect. 5, the label distributions of the scientific test sets are heavily skewed, thus leaving macro-F1 vulnerable to performance shifts on very small classes. We therefore investigated micro-averaged F1-score, as it aggregates the contributions of all classes, without noting class imbalance. In our setting, micro-F1 consistently scores approximately 1% above macro-F1, emphasizing the good results of our models on big classes. The only special case is, as mentioned above, w2v$_{\text{arXiv}}$ on SE18-7$_{\text{noisy}}$, with a macro-F1 of 85.79 and micro-F1 of only 83.94. By contrast, ELMo$_1$ delivers results of 85.63 macro- and 86.20 micro-F1; in other words, similar macro-F1 but

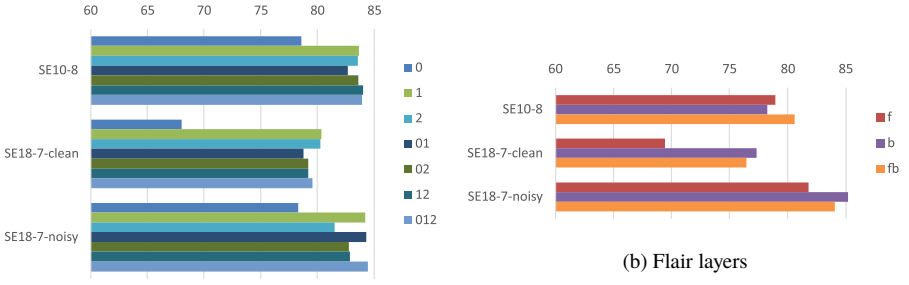
[11] We evaluated combinations of three embeddings as well, but results did not improve.

quite different micro-F1. Hence, the result of $w2v_{arXiv}$ on $SE18\text{-}7_{noisy}$ must be a case of overfitting on small classes and overestimating classifier performance by usage of macro-F1. Note that we still outperform the best previous SVM for all data sets if we add contextual embeddings to ClaiRE.

6.3 Analysis of Contextual Layers

After comparing different word embeddings and embedding types in the previous section, we will now look at the best configuration of context-sensitive embeddings. Both utilised context-sensitive models produce word vectors from the internal states of different LM-layers. As the best combination of ELMo-layers depends on the task at hand [Pe18], we investigate relation classification performance for different configurations of ELMo- and Flair-layers.

As shown in Fig. 2a, the static ELMo-layer $ELMo_0$ performs notably worse for all three datasets, while $ELMo_1$ is the best singular layer. Combining different layers does not change performance for relation classification considerably. For Flair we find that there exists no clear advantage for a layer combination across datasets (cf. Fig. 2b). As the backwards-layer $Flair_b$ alone scores best for two out of three tasks, we chose it for our experiments in Sect. 6.2.



(a) ELMo layers

(b) Flair layers

Fig. 2: Results as macro-F1 for different combinations of layers from context-sensitive embeddings (including lexical features).

6.4 Analysis of Nearest Neighbours

To illustrate the behaviour of different embedding types on two domains, general and scientific, we present nearest neighbours in the embedding space for some ambiguous words in Tab. 6. We determine closeness by means of cosine similarity and report five nearest neighbours of any vocabulary entry in the case of w2v and $w2v_{arXiv}$. We additionally compute the closest word in the ELMo embedding space for a) a token appearance in SE10-8 and b) an appearance in SE18-7 and report their associated sentences.

As shown in Tab. 6, ambiguous word have a different meaning in the general and the scientific domain, as is evidenced by their neighbourhood in the respective domain embeddings. In contrast, ELMo embeds words into a vector space depending on their context, enabling different nearest neighbours matching their word sense. We assume that this distinction between words senses contributes to a mapping of entities to relations.

WE	**word (in context)**	**nearest neighbours**
w2v	tree	trees, pine tree, oak tree, evergreen tree, fir tree
$w2v_{arXiv}$		trees, subtree, subtrees, leaf, graph
ELMo	a) An oak **tree** grows from an acorn.	Winter is here and the little fir **tree** stands lonely in the forest.
	b) We use decision **trees** to learn the controllers.	This paper describes novel and practical Japanese parsers that uses decision **trees**.
w2v	string	spate, slew, rash, litany, flurry
$w2v_{arXiv}$		strings, superstring, worldsheet, brane, worldsheets
ELMo	a) A **string** of pack ponies trotted through the pines behind them.	I remembered about a **string** of rosary beads [. . .]
	b) One is **string** similarity based on edit distance.	We take a selection of both bag of words and segment order sensitive **string** comparison methods [. . .]

Tab. 6: Most similar words for different static embeddings and ELMo.

7 Conclusion

Our intuition was that general word embeddings would fail to capture the meaning of some words for relation classification on scientific data, while specialised word embeddings would in turn fail to work outside their domain. This intuition is supported by our results. We also hypothesised that context-sensitive word embeddings would be able to generalise across domains, as they can model multiple meanings of a word and distinguish them by their current context. This assumption also seems to hold true, as evidenced by the consistently great performance of ELMo and the good performance of Flair.

Overall, ECLaiRE - our combination of ClaiRE and ELMo - a simple rbf-SVM with a few hand-coded features and context-sensitive word embeddings, is able to outperform the best SVM classifiers so far and even achieves similar results to a complex neural network architecture for the $SE18\text{-}7_{clean}$ task. Thus, it may be useful to introduce context-sensitive word embeddings, especially ELMo, to more relation classification datasets from different domains.

Bibliography

[ABV18] Akbik, Alan; Blythe, Duncan; Vollgraf, Roland: Contextual String Embeddings for Sequence Labeling. In: COLING. pp. 1638–1649, 2018.

[BMiK15] Bollegala, Danushka; Maehara, Takanori; ichi Kawarabayashi, Ken: Unsupervised Cross-Domain Word Representation Learning. In: ACL. The Association for Computer Linguistics, pp. 730–740, 2015.

[Bo17] Bojanowski, Piotr; Grave, Edouard; Joulin, Armand; Mikolov, Tomas: Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics, 5:135–146, 2017.

[CCP18] Camacho-Collados, José; Pilehvar, Mohammad Taher: From Word to Sense Embeddings: A Survey on Vector Representations of Meaning. Journal of Artificial Intelligence Research, 63:743–788, 2018.

[Ch13] Chelba, Ciprian; Mikolov, Tomas; Schuster, Mike; Ge, Qi; Brants, Thorsten; Koehn, Phillipp; Robinson, Tony: , One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling, 2013.

[CLS14] Chen, Xinxiong; Liu, Zhiyuan; Sun, Maosong: A unified model for word sense representation and disambiguation. In: EMNLP. pp. 1025–1035, 2014.

[Gá18] Gábor, Kata; Buscaldi, Davide; Schumann, Anne-Kathrin; QasemiZadeh, Behrang; Zargayouna, Haïfa; Charnois, Thierry: SemEval-2018 Task 7: Semantic Relation Extraction and Classification in Scientific Papers. In: SemEval@NAACL-HLT. pp. 679–688, 2018.

[He09] Hendrickx, Iris; Kim, Su Nam; Kozareva, Zornitsa; Nakov, Preslav; Ó Séaghdha, Diarmuid; Padó, Sebastian; Pennacchiotti, Marco; Romano, Lorenza; Szpakowicz, Stan: Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In: Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions. pp. 94–99, 2009.

[He18] Hettinger, Lena; Dallmann, Alexander; Zehe, Albin; Niebler, Thomas; Hotho, Andreas: ClaiRE at SemEval-2018 Task 7: Classification of Relations using Embeddings. In: Proceedings of International Workshop on Semantic Evaluation. 2018.

[HR18] Howard, Jeremy; Ruder, Sebastian: Universal Language Model Fine-tuning for Text Classification. In: ACL. Association for Computational Linguistics, 2018.

[HS97] Hochreiter, Sepp; Schmidhuber, Jürgen: Long Short-Term Memory. Neural Computation, 9(8):1735–1780, November 1997.

[IPN15] Iacobacci, Ignacio; Pilehvar, Mohammad Taher; Navigli, Roberto: Sensembed: Learning sense embeddings for word and relational similarity. In: COLING. volume 1, pp. 95–105, 2015.

[JP15] Johansson, Richard; Pina, Luis Nieto: Embedding a semantic network in a word space. In: NAACL-HLT. pp. 1428–1433, 2015.

[Ki14] Kim, Yoon: Convolutional Neural Networks for Sentence Classification. In: EMNLP. pp. 1746–1751, 2014.

[LL13] Labutov, Igor; Lipson, Hod: Re-embedding words. In: ACL. pp. 489–493, 2013.

[Ma14] Marco Baroni, Georgiana Dinu, Germán Kruszewski: Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. COLING, 1:238–247, 2014.

[Mc17] McCann, Bryan; Bradbury, James; Xiong, Caiming; Socher, Richard: Learned in Translation: Contextualized Word Vectors. In: Advances in Neural Information Processing Systems. 2017.

[Mi13] Mikolov, Tomas; Sutskever, Ilya; Chen, Kai; Corrado, Greg S; Dean, Jeff: Distributed Representations of Words and Phrases and their Compositionality. In: NIPS, pp. 3111–3119. Curran Associates, Inc., 2013.

[Ni17] Niebler, Thomas; Becker, Martin; Pölitz, Christian; Hotho, Andreas: Learning Semantic Relatedness from Human Feedback Using Relative Relatedness Learning. In: ISWC. 2017.

[Pe18] Peters, Matthew E.; Neumann, Mark; Iyyer, Mohit; Gardner, Matt; Clark, Christopher; Lee, Kenton; Zettlemoyer, Luke: Deep Contextualized Word Representations. In: NAACL-HLT. pp. 2227–2237, 2018.

[PSM14] Pennington, Jeffrey; Socher, Richard; Manning, Christopher D: Glove: Global Vectors for Word Representation. In: EMNLP. volume 14, pp. 1532–1543, 2014.

[RH10] Rink, Bryan; Harabagiu, Sanda: Utd: Classifying semantic relations by combining lexical and semantic resources. In: Proceedings of the 5th International Workshop on Semantic Evaluation. pp. 256–259, 2010.

[RHZ18] Rotsztejn, Jonathan; Hollenstein, Nora; Zhang, Ce: , ETH-DS3Lab at SemEval-2018 Task 7: Effectively Combining Recurrent and Convolutional Neural Networks for Relation Classification and Extraction, 2018.

[SCH17] Speer, Robert; Chin, Joshua; Havasi, Catherine: , ConceptNet 5.5: An Open Multilingual Graph of General Knowledge, 2017.

[SP97] Schuster, Mike; Paliwal, Kuldip K: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11):2673–2681, 1997.

[Ta14] Tang, Duyu; Wei, Furu; Yang, Nan; Zhou, Ming; Liu, Ting; Qin, Bing: Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification. In: COLING. pp. 1555–1565, 2014.

[To03] Toutanova, Kristina; Klein, Dan; Manning, Christopher D; Singer, Yoram: Feature-rich part-of-speech tagging with a cyclic dependency network. In: HLT-NAACL. pp. 173–180, 2003.

[Wa16] Wang, Linlin; Cao, Zhu; de Melo, Gerard; Liu, Zhiyuan: Relation Classification via Multi-Level Attention CNNs. In: COLING. volume 1, pp. 1298–1307, 2016.

[WLW04] Wu, Ting-Fan; Lin, Chih-Jen; Weng, Ruby C.: Probability Estimates for Multi-class Classification by Pairwise Coupling. Journal of Machine Learning Research, 5(Aug):975–1005, 2004.

[YLZ17] Yang, Wei; Lu, Wei; Zheng, Vincent: A Simple Regularization-based Algorithm for Learning Cross-Domain Word Embeddings. In: EMNLP. pp. 2898–2904, 2017.

[Yu17] Yu, Liang-Chih; Wang, Jin; Lai, K. Robert; Zhang, Xue-Jie: Refining Word Embeddings for Sentiment Analysis. In: EMNLP. pp. 534–539, 2017.