CNN- and LSTM-based Claim Classification in Online User Comments

Chinnappa Guggilla and **Tristan Miller** and **Iryna Gurevych**
Research Training Group AIPHES
Ubiquitous Knowledge Processing (UKP) Lab
Technische Universität Darmstadt
https://www.aiphes.tu-darmstadt.de/
https://www.ukp.tu-darmstadt.de/

Abstract

When processing arguments in online user interactive discourse, it is often necessary to determine their bases of support. In this paper, we describe a supervised approach, based on deep neural networks, for classifying the claims made in online arguments. We conduct experiments using convolutional neural networks (CNNs) and long short-term memory networks (LSTMs) on two claim data sets compiled from online user comments. Using different types of distributional word embeddings, but without incorporating any rich, expensive set of features, we achieve a significant improvement over the state of the art for one data set (which categorizes arguments as factual vs. emotional), and performance comparable to the state of the art on the other data set (which categorizes propositions according to their verifiability). Our approach has the advantages of using a generalized, simple, and effective methodology that works for claim categorization on different data sets and tasks.

1 Introduction

Argumentation mining is a relatively new subfield in natural language processing that aims to automatically identify and extract arguments, and their underlying structures, from textual documents (Moens et al., 2007; Palau and Moens, 2009; Wyner et al., 2010; Feng and Hirst, 2011; Ashley and Walker, 2013; Stab and Gurevych, 2014). Some such documents are written by professionals and contain well-formed, explicit arguments—i.e., propositions supported by evidence and connected through reasoning. However, informal arguments in online argumentative discourses can exhibit different styles. Recent work has begun to model different aspects of these naturally occurring lay arguments, with tasks including stance classification (Somasundaran and Wiebe, 2009; Walker et al., 2012), argument summarization (Misra et al., 2015), sarcasm detection (Justo et al., 2014) and classification of propositions and arguments (Park and Cardie, 2014; Park et al., 2015; Oraby et al., 2015). Of particular interest is the fact that arguments in online user comments, unlike those written by professionals, often have inappropriate or missing justifications. Recognizing such propositions and determining the appropriate types of support can be useful for assessing the strength of the supporting information and, in turn, the strength of the whole argument.

To this end, two previous studies have produced data sets and methods for classifying propositions in online argumentative discourse. The first of these studies (Park and Cardie, 2014) compiled online user comments from a discussion website and developed a framework for automatically classifying each proposition as either “unverifiable”, “verifiable non-experiential”, or “verifiable experiential”, where the appropriate types of support are reason, evidence, and optional evidence, respectively. The second study, Oraby et al. (2015), uses a different online corpus (Walker et al., 2012) of short argumentative responses to quotes, and classifies each response as either “factual” or “feeling” according to whether the support invoked appeals to facts or to emotions. In this paper, we use the term “claim” loosely to refer to an individual proposition (a sentence or independent clause) in an argument, or to a short argumentative text containing one or more propositions.

In classifying propositions, Park and Cardie (2014) followed previous work such as Reed et al. (2008) and Palau and Moens (2009), employing supervised learning methods. Despite using a rich set of linguistic features, these approaches suffer from low accuracy. Moreover, generating these features can be a tedious and complex process. In this paper, we show that state-of-the-art performance in claim classification for online user comments can be achieved without the need for expensive features. Our work, which employs CNN- and LSTM-based deep neural networks, is inspired by advances in sentence classification (Kim, 2014) and sequence classification (Hochreiter and Schmidhuber, 1997) using distributional word representations and deep learning. In particular, our approach leverages word2vec[1] distributional embeddings, dependency context–based embeddings (Levy and Goldberg, 2014), and factuality/certainty-indicating embeddings for improving claim classification. (We refer to these embeddings as *linguistic embeddings*, as these are compiled from linguistic annotations such as dependency relations, verb modalities, and actuality information.) In this paper, we separately evaluate the usefulness of word and linguistic embeddings in the claim classification task on both the aforementioned data sets. We also concatenate (stack) these embeddings and show how these stacked embeddings, as well as tuning of the hyper-parameters, further improves claim classification performance.

2 Background

In this section, we introduce two main approaches for claim classification: feature-rich supervised learning and distributional word embeddings. We then discuss the recent use of convolutional neural networks and long short-term memory networks in the related task of sentence classification.

2.1 Methods Based on a Rich Set of Features

Oraby et al. (2015) classify arguments as emotional or factual using a set of linguistic patterns extracted from unlabelled arguments, provided the argument matches at least three patterns in the category. Although this approach has good precision, its recall is significantly lower than that of a supervised unigram baseline using Bayesian classifier.

Park and Cardie (2014) classify propositions as verifiable non-experiential, verifiable experiential, or unverifiable using a support vector machine (SVM). The classifier employs a rich set of features including *n*-grams, part of speech tags, imperative expressions, speech events, emotions, sentiment, person, and tense. Though this approach classifies unverifiable statements reasonably well, its performance on the two classes of verifiable propositions is low (44–70% F_1). In light of the observation that certain types of propositions tend to occur together, Park et al. (2015) propose an intuitive extension to this approach, framing the proposition classification task as a sequence-labelling problem. This extended approach employs conditional random fields (CRF) using dictionary-based features along with all the features from the original technique. However, it resulted in lower accuracy than the SVMs.

Ferreira and Vlachos (2016) addressed the task of determining the stance of news article headlines with respect to claims from a data set of rumours. The authors used a logistic regression classifier using various features, such as bag of words, paraphrase entailment alignment scores, and word2vec embedding features, that examine the headline and its agreement with the claim. The work in this paper is focused on stance classification but the claims in the data set are related to the data sets used in our work.

2.2 Distributional Word Embeddings

Traditional supervised learning approaches to NLP tasks depend heavily on manual annotation, and often suffer from data sparseness. Distributional representations of words, also known as word embeddings, can be learned from large, unlabelled corpora using neural networks, and encode both syntactic and semantic properties of words. Studies have found the learned word vectors to capture linguistic regularities and to collapse similar words into groups (Mikolov et al., 2013b). Their utility in tasks such as sentiment classification (Kim, 2014) is well attested.

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

Dependency-based Embeddings. Claims containing multiple clauses or propositions might be better distinguished with the help of dependency embeddings inferred from the respective proposition contexts. Consider the following claim from one of our data sets: "The Governor said that he enjoyed it." In this claim, the main clause, "The Governor said", is the core proposition, which excludes consideration of the remainder. The reason is that "said" is a reporting predicate, so it is unnecessary to verify whether or not the governor really has enjoyed the object mentioned in the subordinate clause. In some other claims, it is the subordinate rather than the main clause predicate that decides the claim type. Park and Cardie (2014) extracted clause-specific features using the Stanford syntactic parser and the Penn Treebank. (Merely using clause tags without capturing dependencies for important clauses may not help much in distinguishing objective verifiable claims from unverifiable subjective ones.) Park and Cardie (2014) also used tense and person counts for distinguishing verifiable claims from unverifiable claims. We hypothesize that word2vec and dependency context–based embeddings can inherently capture these linguistic characteristics and can replace these features. Dependency context based embeddings capture functional similarities across the words using different contexts (Levy and Goldberg, 2014). Komninos and Manandhar (2016) have shown that dependency-based models produce word embeddings that better capture functional properties of words for question type classification and relation detection.

Task-specific Embeddings. Compiling embeddings for the specific vocabulary present in the task data can also be helpful in a classification task. Tang et al. (2014) use enriched task-specific word embeddings and show improvement in a Twitter sentiment classification task. Park and Cardie (2014) compiled a speech-event lexicon containing the most frequent speech anchors (predicates such as "said" and "wrote") from MPQA 2.0, a corpus manually annotated for opinions and other private states. These anchors can help in correctly distinguishing verifiable claims from unverifiable ones when the propositions contain both objective and subjective expressions. In our work, we use factual embeddings learned from the labelled FactBank corpus (Saurí and Pustejovsky, 2009) containing various speech event predicates (see §3.3). Such factual embeddings could help in resolving various predicate ambiguities present in the argumentative propositions.

2.3 Deep Neural Networks for Text Classification

Deep neural networks, with or without word embeddings, have recently shown significant improvements over traditional machine learning–based approaches when applied to various sentence- and document-level classification tasks.

Kim (2014) have shown that CNNs outperform traditional machine learning–based approaches on several tasks, such as sentiment classification, question type classification, and subjectivity classification, using simple static word embeddings and tuning of hyper-parameters. Zhang et al. (2015) proposed character-level CNNs for text classification. Lai et al. (2015) and Visin et al. (2015) proposed recurrent CNNs, while Johnson and Zhang (2015) proposed semi-supervised CNNs for solving a text classification task. Tang et al. (2015) used a document classification approach based on recurrent neural networks (RNNs) and showed an improvement on a sentiment classification task. Palangi et al. (2016) proposed sentence embedding using an LSTM network for an information retrieval task. Zhou et al. (2016) proposed attention-based, bidirectional LSTM networks for a relation classification task. Augenstein et al. (2016) employed a weakly supervised conditional LSTM encoding approach to stance detection for unseen targets on Twitter stance detection data, and presented improved results. RNNs model text sequences effectively by capturing long-range dependencies among the words. LSTM-based approaches based on RNNs effectively capture the sequences in the sentences when compared to the CNN and SVM-based approaches.

3 Claim Classification

Here we present two deep learning–based methods for claim classification, the first of which uses CNNs and the second of which uses LSTMs. In §3.3, we also show how different pre-trained distributional linguistic embeddings are incorporated into CNNs and LSTMs to improve the classification results.

3.1 CNN-based Claim Classification

Collobert et al. (2011) adapted the original CNN proposed by LeCun and Bengio (1995) for modelling natural language sentences. Following Kim (2014), we present a variant of the CNN architecture with four layer types: an input layer, a convolution layer, a max pooling layer, and a fully connected softmax layer. Each claim in the input layer is represented as a sentence comprised of distributional word embeddings. Let $\vec{v}_i \in \mathbb{R}^k$ be the k-dimensional word vector corresponding to the ith word in the sentence. Then a sentence S of length ℓ is represented as the concatenation of its word vectors:

$$S = \vec{v}_1 \oplus \vec{v}_2 \oplus \cdots \oplus \vec{v}_\ell. \tag{1}$$

Word2vec embeddings which are learned using the bag-of-words representation of the contexts yield broad topical similarities, while using dependency-based contexts yields more functional similarities (Levy et al., 2015). In addition, with word2vec (E) embeddings, we use linguistically motivated pre-trained dependency embeddings (D) and task-specific factual embeddings (F) for capturing syntactic and functional regularities encoded in the propositions, in order to better distinguish different types of claims.

To incorporate these linguistic embeddings at word level into the learning process, we extend the network as illustrated in Figure 1a. Inspired by Baroni et al. (2012)'s supervised distributional concatenation method and a linguistically informed CNN (Ebert et al., 2015), we concatenate word2vec (E), dependency (D), and factual (F) word embeddings corresponding to the ith input word into a merged vector $\vec{c}_i \in \mathbb{R}^{k+m+n}$:

$$\vec{c}_i = [\vec{e}_i, \vec{d}_i, \vec{f}_i], \tag{2}$$

where $\vec{e}_i$, $\vec{d}_i$, and $\vec{f}_i$ represent, respectively, the concatenated word2vec, dependency, and factual embeddings corresponding to i_{th} word in the sentence. In the final representation, every input claim from the data set is represented using combined word2vec and linguistic embeddings in the network as in Equation 1, where each $\vec{v}_i = \vec{c}_i$.

In the convolution layer, for a given word sequence within a claim, a convolutional word filter P is defined. Then, the filter P is applied to each word in the sentence to produce a new set of features. We use a non-linear activation function such as rectified linear unit (ReLU) for the convolution process and max-over-time pooling (Collobert et al., 2011; Kim, 2014) at pooling layer to deal with the variable claim size. After a series of convolutions with different filters with different heights, the most important features are generated. Then, this feature representation, Z, is passed to a fully connected penultimate layer and outputs a distribution over different labels:

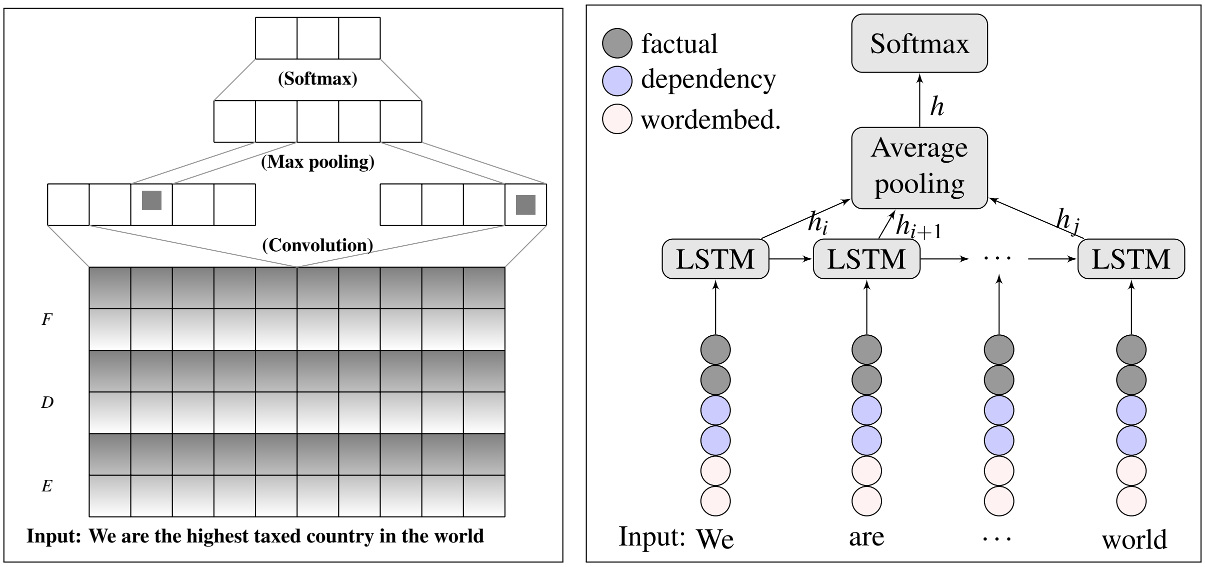
$$y = \text{softmax}(W \cdot Z + b), \tag{3}$$

where y denotes a distribution over different claims labels, W is the weight vector learned from the stacked representation of all embeddings from the training corpus, and b is the bias term.

3.2 LSTM-based Claim Classification

In case of CNN, concatenating words with various window sizes, works as n-gram models but do not capture long-distance word dependencies with shorter window sizes. A larger window size can be used, but this may lead to data sparsity problem. In order to encode long-distance word dependencies, we use long short-term memory networks, which are a special kind of RNN capable of learning long-distance dependencies. LSTMs were introduced by Hochreiter and Schmidhuber (1997) in order to mitigate the vanishing gradient problem (Gers et al., 2000; Gers, 2001; Graves, 2013; Pascanu et al., 2013).

The model illustrated in Figure 1b is composed of a single LSTM layer followed by an average pooling and a softmax regression layer. Each claim is represented as a sentence (S) in the input layer. Thus, from an input sequence, $S_{i,j}$, the memory cells in the LSTM layer produce a representation sequence $h_i, h_{i+1}, \ldots, h_j$. This representation sequence is then averaged over all time steps, resulting in a final feature representation h. Finally, this representation is fed to a logistic regression layer to predict the claim labels for unseen input claims.



(a) CNN with multiple word embeddings

(b) LSTM with multiple word embeddings

Figure 1: Illustration of two methods for claim classification

[There *always* **seems** to be some other amount] [I *later* **must** pay]
predicate (possibility) *modal (obligation)*

Figure 2: Example verifiable non-experiential claim with signals indicating factuality and certainity

As with the CNN architecture shown in the previous section, for each claim, we encode word2vec, dependency, and factual embeddings in the input layer into a variation of the standard LSTM network. As our results demonstrate, the LSTM encoder can effectively capture informative features from the concatenated embedding representation and classify different types of argumentative claims.

3.3 Word Embeddings

In order to better capture the syntactic contexts of words and the factuality indicators of propositions, we employ two linguistically motivated word embeddings in addition to the usual word2vec ones: dependency-based embeddings, and factuality- and certainty-signalling emeddings.

Word2vec Embeddings. We use word embeddings from word2vec which are learned using the skip-gram model of Mikolov et. al (2013a,b) by predicting linear context words surrounding the target words. These word vectors are trained on about 100 billion words from a Google News corpus. As word embeddings alone have shown good performance in various classification tasks, we also use them in isolation, with varying dimensions, in our CNN and LSTM experiments. In the case of CNN, a word embedding size of 300, together with other network parameters, resulted in high accuracy on the claim verifiability data set. In the case of LSTM, word embeddings of size 300 also produced good accuracy on the claim verifiability data set.

Dependency-based Word Embeddings. We use Levy and Goldberg's (2014) dependency-based word embeddings in our claim classification task. These embeddings are learned using dependency-based contexts from an English Wikipedia corpus containing about 175 000 words and over 900 000 distinct syntactic contexts. Dependency-based embeddings are encoded in the input layers of both our CNN and LSTM, as shown in Figure 1. Dependency embeddings of size 100 are concatenated with equally sized word2vec and factual embeddings, resulting in a 300-dimension concatenated embedding vector.

Factuality- and Certainty-signalling Embeddings. We investigate the use of certainty- and factuality-related distributed signals for distinguishing claims. In online argumentative discourse, claims often

	Factual	Feeling	Total
Train	2426	1667	4093
Test	347	239	586
Total	2773	1906	4679

Table 1: Data splits (factual/feeling data set)

	Ver. exp.	Ver. non-exp.	Unver.	Total
Train	900	987	4459	6346
Test	367	370	1687	2424
Total	1267	1357	6146	8770

Table 2: Data splits (verifiability data set)

serve as implicit arguments with inappropriate or missing justification (Park and Cardie, 2014). The certainty and factuality signals present in such claims may be appropriate for determining its factuality or verifiability. As the claims in our data set are objective, subjective and factual types, predicates, adverbs and other modals (related to certainty and factuality) present in FactBank 1.0 may help in better distinguishing various types of claims.

As an example, consider the sentence in Figure 2, a complex claim of type "verifiable non-experiential". The predicate "seems" and the modal verb "must" can be viewed as certainty and factuality information related to the speaker's commitment to their utterance. Factual embeddings of these co-occurrence indicators can help in better identifying the type of the claim. We compile these extra linguistic factual and certainty signals from FactBank (Saurí and Pustejovsky, 2009), a corpus annotated with factuality and certainty indicators very much similar to the word2vec embeddings. These annotations are basically related to certainty, possibility, and probability, with positive and negative polarities. We used the gensim (Řehůřek and Sojka, 2010) word2vec program to compile embeddings from FactBank. We compiled 300-dimensional factual embedding vectors for the words that appear at least five times in FactBank, and for rest of the vocabulary, embedding vectors are assigned uniform distribution in the range of $[-0.25, 0.25]$. In our CNN and LSTM experiments, we integrate factual embeddings (denoted by F above). We also concatenate factual embeddings with other dependency and word embeddings, as shown in Figure 1.

4 Data Sets and Experimental Setup

4.1 Data Sets

Our experiments use the two claim data sets introduced in §1, further details of which are given below.

Factual and Feeling Debate Forum Posts (Walker et al., 2012). This corpus is compiled from the Internet Argument Corpus. It consists of quote–response pairs that are manually annotated according to whether the response is primarily a "factual"- or "feeling"-based argument. In our experiments, we use the training and test splits from Oraby et al. (2015); these consist of claims that can span multiple sentences. The annotation distribution for these splits is shown in Table 1. We also use a development set to tune the hyper-parameters of the model.

Verifiable and Unverifiable User Comments (Park and Cardie, 2014). This corpus consists of 9476 manually annotated sentences and independent clauses from 1047 user comments extracted from the Regulation Room website.[2] Park and Cardie (2014) and Park et al. (2015) used this corpus for examining each proposition with respect to its verifiability to determine the desirable types of support for the analysis of arguments. The propositions are manually annotated with three classes—"verifiable experiential", "verifiable non-experiential", and "unverifiable"—where the support types are evidence, optional evidence, and reason, respectively. The annotation distribution and our train/test splits are shown in Table 2.

4.2 Experimental Setup

We model claim classification as a sentence classification task. We perform binary classification on the factual/feeling data set, and multi-class classification on the verifiability data set. We used Kim's (2014) Theano implementation of CNN for training the CNN model and a variant of the standard Theano implementation[3] for training the LSTM network. We initialized the word2vec, dependency, and factual

[2] http://www.regulationroom.org/

[3] http://deeplearning.net/tutorial/lstm.html

embeddings in both the CNN and LSTM models. Unknown words from the pre-compiled embeddings were initialized randomly in the range $[-0.25, 0.25]$. We updated all three embedding vectors during the training. We also produced a stacked embedding where all three types of embeddings, with dimensionality 100, were concatenated. In the CNN approach, we used a stochastic gradient descent–based optimization method for minimizing the cross entropy loss during the training with the Rectified Linear Unit (ReLU) non-linear activation function. Window filter sizes were set at $[3, 4, 5]$. In the case of LSTM, model was trained using an adaptive learning rate optimizer, ADADELTA (Zeiler, 2012), over shuffled mini-batches with the sigmoid activation function at input, output and forget gates, and the tanh non-linear activation function at cell state.

Tuning Hyper-parameters. We manually explored hyper-parameters such as drop-out (for avoiding over-fitting), and batch size and learning rates (for improving performance) on development sets of both data sets. We performed tuning on the verifiability development data set obtained by splitting the corpus into an 85% training set and a 15% development set. We tuned the hyper-parameters on a 20% development set obtained from Oraby et al. (2015) on the factual vs. feeling data set. We varied batch sizes (12–64), drop-out (0.1–0.6), embedding sizes (50–300), and learning rate (0.0001–0.001) on both data sets and across all embeddings. We obtained the best CNN performance with learning rate decay 0.95, batch size 50, drop-out 0.5, and embedding size 300. For LSTM, we got the best results with learning rate 0.001, drop-out 0.5, and embedding size 300 for both data sets; the optimal batch size was 24 for the verifiability data set but 32 for the factual vs. feeling data set.

SVM Classification on the Factual vs. Feeling Data Set. SVM classifiers find the hyperplane that best discriminates between positive and negative instances (Cristianini and Shawe-Taylor, 2000). We used the SVM classifier SMO (Hall et al., 2009) from the DKPro TC framework (Daxenberger et al., 2014) for factual vs. feeling claims classification. Surface-level top *k* *n*-grams are used as features for building the model. We used uni-, bi-, and trigrams, and varied *k* from 500 to 5000. We obtained the best results with the top 500 *n*-gram features.

5 Results and Analysis

We compare our methods with several state-of-the-art methods for claim classification, as described in §2. In these tables, the highest accuracy values for precision, recall and F_1 measure are specified in bold font.

Verifiability Data Set. Park and Cardie (2014) and Park et al. (2015) performed claim classification on this data set using SVM and CRF classifiers. The former classifier was found to yield better results. Both approaches employed various lexical and shallow semantic features. The authors also report baseline results using simple unigram features. We considered the SVM-based results[4] a baseline for comparison with ours. The results of our own experiments on the same data set, using CNN and LSTM methods together with the various embeddings mentioned in §3.3, are shown in Table 3. We macro-averaged F_1 across all the classes. Using word embeddings alone in the CNN method, our results (70.47%) were comparable to those of the SVM (68.99%) and exceeded those of the CRF method (63.63%). In a concatenated embeddings setting, CNN achieves 70.34% F_1. The LSTM performance is low when compared to the CNN approach, but comparable to the SVM-based approach. LSTM also performed better than the sequential CRF baseline.

We computed train, validation, and test error rates with respect to the number of epochs during training for the CNN and LSTM approaches. In the case of LSTM, the best classification accuracy is obtained between 5 and 12 epochs, and in case of CNN, at between 5 and 20 epochs. Confusion matrices showing the assignments of our best-performing LSTM and CNN classifiers are shown in Tables 4 and 5, respectively. Both classifiers show a similar pattern of errors. Verifiable experiential and non-experiential claims were not confused as much with each other as they were with unverifiable claims; this may be an artifact of the latter being the majority class. When unverifiable claims were misclassified, they were more

[4] Results are evaluated in a one-vs.-all binary classification setting.

		Unverifiable			Verifiable non-exp.			Verifiable exp.			Macro
System	**Features**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**avg. F_1**
Rand.		71.28	69.59	70.42	15.13	15.13	15.13	15.26	15.26	15.26	33.65
SVM	feat.-rich	82.14	89.69	85.75	51.67	37.57	43.51	73.48	62.67	67.65	65.63
SVM	unigram	**86.86**	83.05	84.91	49.88	**55.14**	52.37	66.67	**73.02**	69.70	68.99
CRF		80.35	**93.30**	84.91	**60.34**	28.38	38.60	**74.57**	59.13	65.96	63.63
CNN	word2vec	85.74	88.74	**87.21**	57.19	49.46	53.04	72.07	70.30	71.17	**70.47**
	dep. embed.	86.46	85.95	86.21	55.86	54.05	**54.94**	67.09	71.12	69.05	70.06
	fact. embed.	83.65	87.01	85.30	55.10	43.78	48.79	64.00	65.39	64.69	66.26
	all embed.	85.75	88.14	86.93	54.87	50.27	52.47	67.14	77.38	**71.90**	70.34
LSTM	word2vec	84.86	81.09	82.93	42.66	51.08	46.49	67.21	67.58	67.39	65.60
	dep. embed.	83.31	85.83	84.55	46.09	46.21	46.15	72.38	62.12	66.86	65.85
	fact. embed.	84.50	85.65	85.07	51.12	42.97	46.70	64.02	70.30	67.01	66.26
	all embed.	84.63	82.27	83.44	42.91	54.86	48.16	72.67	61.58	66.67	66.09

Table 3: Classifier performance on the verifiability data set. The SVM and CRF classifiers are those from Park and Cardie (2014); "Rand." is the random baseline.

		Predicted		
		Ver. exp.	**Ver. non-exp.**	**Unver.**
Actual	**Ver. exp.**	258	25	84
	Ver. non-exp.	30	159	181
	Unver.	115	127	1445

Table 4: Confusion matrix for LSTM with factual embeddings (verifiability data set)

		Predicted		
		Ver. exp.	**Ver. non-exp.**	**Unver.**
Actual	**Ver. exp.**	258	19	90
	Ver. non-exp.	28	183	159
	Unver.	72	118	1497

Table 5: Confusion matrix for CNN with word2vec (verifiability data set)

likely to be labelled as verifiable non-experiential, suggesting that the vocabulary employed in the two classes of claims is similar.

Factual vs. Feeling Claims Data Set. In this data set, claims can span more than one sentence, but we treat these as single sentences for the purposes of our experiments. Oraby et al. (2015) performed unsupervised claim classification on this data set using bootstrapped patterns from both unlabelled and labelled data and report accuracy (F_1) of 41.41%. They also report an F_1 of 64.98% for a naïve Bayes supervised classifier using simple unigram and binary features. The focus of their experiment was to discover more factual- and feeling-related patterns from the unlabelled corpus using a small amount of labelled data. In our experiments, both the CNN (79.56% F_1) and the LSTM-based (75.10% F_1) methods using distributional embeddings show significant improvements over the naïve Bayes and SVM-based approaches as shown in Table 6. CNN achieved good accuracy in all embeddings setting. Sequential LSTM's performance is not better than the CNN approach, but LSTM together with word2vec and factual embeddings performed better on this data set.

Confusion matrices for our best LSTM and CNN classifiers are shown in Tables 7 and 8, respectively. We manually examined those factual claims misclassified as feeling and found that they contained a relatively high proportion of personal pronouns, wh-questions, and negations. While these vocabulary terms are typically associated with feeling claims, they are missing from the factuality embeddings learned from FactBank. By contrast, when feeling claims were misclassified as factual, we found that they tend to contain several distinct propositions or clauses, only one of which was emotional in nature. Properly handling these type of claims would require modelling them with intrapropositional relations.

6 Conclusion and Future Work

In this paper, we presented LSTM- and CNN-based deep neural network methods leverging word2vec and linguistic embeddings, and applied these to argumentative claim classification on two data sets.

On the data set of verifiable and unverifiable claims, our CNN approach using word2vec and concatenated embeddings has shown results comparable to those of a state-of-the-art, feature-rich, SVM-based

		Factual			Feeling			
System	**Features**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**Macro avg. F_1**
Random baseline		59.08	59.08	59.08	40.59	40.59	40.59	49.83
Oraby et al. (2015)	patterns	79.9	40.1	53.4	63.0	19.2	29.4	41.4
Naïve Bayes	unigrams, binary	73.0	67.0	69.8	57.0	65.0	60.7	65.0
SVM	unigrams	76.14	74.86	75.47	64.31	65.81	65.01	70.24
CNN	word2vec	**82.58**	**84.72**	**83.64**	**76.96**	**74.06**	**75.48**	**79.56**
	dep. embed.	78.49	77.81	78.14	68.18	69.04	68.61	73.38
	fact. embed.	76.24	74.93	75.58	64.49	66.12	65.29	70.43
	all embed.	81.98	81.27	81.62	73.14	**74.06**	73.60	77.61
LSTM	word2vec	80.60	77.81	79.18	69.32	72.80	71.02	75.10
	dep. embed.	78.70	76.66	77.66	67.34	69.87	68.58	73.12
	fact. embed.	78.77	81.27	80.00	71.49	68.20	69.81	74.90
	all embed.	77.09	82.42	79.66	71.63	64.43	67.84	73.75

Table 6: Classifier performance on the factual vs. feeling data set.

		Predicted	
		factual	**feeling**
Actual	**factual**	270	77
	feeling	65	174

Table 7: Confusion matrix for LSTM with word2vec (factual vs. feeling data set)

		Predicted	
		factual	**feeling**
Actual	**factual**	294	53
	feeling	62	177

Table 8: Confusion matrix for CNN with word2vec (factual vs. feeling data set)

method. When using an LSTM-based method, the accuracy was somewhat lower, but still better than a CRF. In this case, however, the concatenated embeddings were not any better than the individual ones. On the factual vs. feeling data set, our CNN-based method using word2vec and linguistic embeddings showed good improvements (over 14 percentage points in F_1) over the state-of-the-art Bayes classifier and a 9-point improvement over the SVM baseline, while the LSTM-based method using word2vec and factual embeddings yielded a 10-point improvement over the Bayes classifier and a 5-point improvement over SVM. The LSTM-based method using word2vec and factual embeddings performed better than using other embeddings. We also observed that the performance of sequential LSTM is lower than the CNN but better than the SVM baseline and the sequential CRF method described in prior work.

Our methods are simpler than those described in prior work, and we have demonstrated that they generalize well across claim data sets. Our framework can also be easily adapted to other stacked embeddings to perform various sentence- and document-level classification tasks. In future work, we plan to investigate usage of richer linguistic embeddings, such as factual and word sense embeddings compiled from a larger corpus. We may also consider incorporating inter-proposition predicate relations.

Acknowledgments

This work was funded through the research training group Adaptive Preparation of Information from Heterogeneous Sources (AIPHES, GRK 1994/1) and through the German Research Foundation (DFG).

References

Kevin D. Ashley and Vern R. Walker. 2013. From information retrieval (IR) to argument retrieval (AR) for legal cases: Report on a baseline study. In Kevin D. Ashley, editor, *Legal Knowledge and Information Systems*, volume 259 of *Frontiers in Artificial Intelligence and Applications*, pages 29–38. IOS Press.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. *CoRR*, abs/1606.05464.

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-chieh Shan. 2012. Entailment above the word level in distributional semantics. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 23–32. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Nello Cristianini and John Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, New York, NY.

Johannes Daxenberger, Oliver Ferschke, Iryna Gurevych, and Torsten Zesch. 2014. DKPro TC: a Java-based framework for supervised learning experiments on textual data. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 61–66, June.

Sebastian Ebert, Ngoc Thang Vu, and Hinrich Schütze. 2015. A linguistically informed convolutional neural network. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 109–114.

Vanessa Wei Feng and Graeme Hirst. 2011. Classifying arguments by scheme. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 987–996.

William Ferreira and Andreas Vlachos. 2016. Emergent: A novel data-set for stance classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1168, June.

Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10):2451–2471.

Felix Gers. 2001. *Long Short-term Memory in Recurrent Neural Networks*. Ph.D. thesis, Universität Hannover.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Rie Johnson and Tong Zhang. 2015. Semi-supervised convolutional neural networks for text categorization via region embedding. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 919–927.

Raquel Justo, Thomas Corcoran, Stephanie M. Lukin, Marilyn Walker, and M. Inés Torres. 2014. Extracting relevant knowledge for the detection of sarcasm and nastiness in the social web. *Knowledge-Based Systems*, 69:124–133.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Alexandros Komninos and Suresh Manandhar. 2016. Dependency based embeddings for sentence classification tasks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1490–1500.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2267–2273.

Yann LeCun and Yoshua Bengio. 1995. Convolutional networks for images, speech, and time series. In Michael A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 255–258. MIT Press, Cambridge, MA.

Omer Levy and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 302–308.

Omer Levy, Steffen Remus, Chris Biemann, Ido Dagan, and Israel Ramat-Gan. 2015. Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

Amita Misra, Pranav Anand, Jean E. Fox Tree, and Marilyn Walker. 2015. Using summarization to discover argument facets in dialog. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 430–440.

Marie-Francine Moens, Erik Boiy, Raquel Mochales Palau, and Chris Reed. 2007. Automatic detection of arguments in legal texts. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law*, pages 225–230.

Shereen Oraby, Lena Reed, Ryan Compton, Ellen Riloff, Marilyn Walker, and Steve Whittaker. 2015. And that's a fact: Distinguishing factual and emotional argumentation in online dialogue. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 116–126.

Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):694–707.

Raquel Mochales Palau and Marie-Francine Moens. 2009. Argumentation mining: The detection, classification and structure of arguments in text. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 98–107.

Joonsuk Park and Claire Cardie. 2014. Identifying appropriate support for propositions in online user comments. In *Proceedings of the 1st Workshop on Argumentation Mining*, pages 29–38.

Joonsuk Park, Arzoo Katiyar, and Bishan Yang. 2015. Conditional random fields for identifying appropriate types of support for propositions in online user comments. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 39–44.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 3, pages 1310–1318.

Chris Reed, Raquel Mochales Palau, Glenn Rowe, and Marie-Francine Moens. 2008. Language resources for studying argument. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, pages 91–100.

Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*.

Roser Saurí and James Pustejovsky. 2009. FactBank: A corpus annotated with event factuality. *Language Resources and Evaluation*, 43(3):227–268.

Swapna Somasundaran and Janyce Wiebe. 2009. Recognizing stances in online debates. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation for Natural Language Processing*, volume 1, pages 226–234.

Christian Stab and Iryna Gurevych. 2014. Identifying argumentative discourse structures in persuasive essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 46–56.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for Twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1555–1565.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432.

Francesco Visin, Kyle Kastner, Kyunghyun Cho, Matteo Matteucci, Aaron C. Courville, and Yoshua Bengio. 2015. ReNet: A recurrent neural network based alternative to convolutional networks. *CoRR*, abs/1505.00393.

Marilyn A. Walker, Jean E. Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012. A corpus for research on deliberation and debate. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 812–817.

Adam Wyner, Raquel Mochales-Palau, Marie-Francine Moens, and David Milward. 2010. Approaches to text mining arguments from legal cases. In Enrico Francesconi, Simonetta Montemagni, Wim Peters, and Daniela Tiscornia, editors, *Semantic Processing of Legal Texts*, volume 6036 of *Lecture Notes in Artificial Intelligence*, pages 60–79. Springer.

Matthew D. Zeiler. 2012. ADADELTA: An adaptive learning rate method. *CoRR*, abs/1212.5701.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 649–657.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 207–212.