HCS at SemEval-2017 Task 5: Sentiment Detection in Business News Using Convolutional Neural Networks

Lidia Pivovarova **Llorenç Escoter** **Arto Klami** **Roman Yangarber**
Department of Computer Science
University of Helsinki
Finland
first.last@cs.helsinki.fi

Abstract

Task 5 of SemEval-2017 involves fine-grained sentiment analysis on financial microblogs and news. Our solution for determining the sentiment score extends an earlier convolutional neural network for sentiment analysis in several ways. We explicitly encode a *focus* on a particular company, we apply a data augmentation scheme, and use a larger data collection to complement the small training data provided by the task organizers. The best results were achieved by training a model on an external dataset and then tuning it using the provided training dataset.

1 Introduction

This paper describes our approach to Task 5 of the SemEval-2017 Challenge—fine-grained sentiment analysis on financial microblogs and news. The task is to determine the *sentiment score* (positive or negative) of a mention of a given company in a business-related text document—a microblog message (Track 1) or a news headline (Track 2).

Our solution, "HCS," is a convolutional neural network to classify sentiment scores. The model's input takes two kinds of information: an article text, a list of *focus* points—positions in the text where a given company is mentioned. Foci allow the model to distinguish company mentions within the text, and to assign different scores to them.

The data provided by the task organisers, (Handschuh et al., 2016), is short, one-sentence messages, with a given focus company. To train the model on additional data, we use the Named Entity (NE) recognition module of PULS (Yangarber and Steinberger, 2009; Huttunen et al., 2013; Atkinson et al., 2011), a news monitoring system, to find company mentions in arbitrary text.

2 Data

The SemEval training set contains 1700 sentences for the microblog track and 1300 news headlines for the headline track, which is a very limited resource for training flexible models. To compensate for the small size of the provided training sets, we built an extended training set. The PULS news monitoring system[1] collects articles from a range of sources of business news (Pivovarova et al., 2013; Du et al., 2016). One of our data sources is a collection of news summaries written by business analysts, which contain metadata annotations.

The metadata does not include sentiment scores. However, the metadata does provide labels that indicate business *events* mentioned in the article, e.g., *Investment*, *Fraud* or *Merger*. The labels are not mutually exclusive, and some documents may have more than one label. There are approximately 300 labels, some of which imply—or weakly imply—positive or negative sentiment. However, most labels do not. We selected only those labels with the most clear sentiment implications: e.g. *Investment*, *New Product*, *Sponsorship*, etc., are considered "positive," while *Fraud*, *Layoff*, *Bankruptcy*, etc., are considered "negative." In total, we used 26 positive and 12 negative labels.

Using these labels, we collected a training set from the corpus of short articles. We selected only documents for which we can infer a clear sentiment score; if a document has event several labels with conflicting sentiment, it is not used for training. Further, we used only those documents, whose headline and first sentence mention exactly one company. The rationale for this is that two companies mentioned together may have different scores. Since our event labels do not provide such detailed information, we avoid these cases to keep the training data as clean as possible. A positive

[1] http://puls.cs.helsinki.fi

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 842–846,
Vancouver, Canada, August 3 - 4, 2017. ©2017 Association for Computational Linguistics

label is considered to have a score of 1 and a negative label is -1.

The dataset produced in this fashion is highly skewed: 90% of the data are positive. We apply a *random undersampling* strategy (Stamatatos, 2008; Erenel and Altınçay, 2013) by randomly selecting a subset of positive documents so that positive and negative training data are more balanced. In our corpus, 100,000 documents have a negative label and mention exactly one company. Thus, the total dataset consists of 200,000 documents. Of these, 10% are used as a development set to determine when to stop training.

3 Approach

Our model is based on a convolutional neural network (Kim, 2014), which demonstrated state-of-the-art performance on sentiment analysis (Tai et al., 2015). The original model is relatively simple, and we adapt it for determining sentiment score for a given company. We add an indicator of *focus* to the input, i.e., the position of the company of interest, for which we wish to determine a sentiment score. We also augment the network by incorporating additional convolutional layers.

An overview of our model is shown in Figure 1. The inputs are fed into the network as zero-padded sentences of a fixed size, where each word is represented as a fixed-dimensional embedding, complemented with a scalar indicator of focus. The inputs are fed into a layer of convolutional filters with multiple widths, optionally followed by deeper convolutional layers. The results of the last convolutional layer are max-pooled, producing a vector with one scalar per filter, which is then fed into a fully-connected layer with dropout regularisation, and a soft-max output layer. The output is a 2-dimensional vector that is interpreted as probability distributions over two possible outcomes: positive and negative. Thus, if an instance has a sentiment score -1 it is mapped into [1, 0], a score of 1 is mapped into [0,1]. A cross-entropy loss function is computed between the network's output and the true value to update the network weights via back-propagation.

Next, we briefly describe the details of the components of the model.

Embeddings: Words are represented by 128-dimensional embeddings. The initial embeddings were trained using GloVe (Pennington et al., 2014) on a corpus of 5 million business news articles. Each document was pre-processed using lemmatisation and named entity (NE) recognition. All NEs of a certain type are mapped to the same token, e.g., all company names have the same embedding.

Following the suggestion of Kim (2014), we tune the embeddings during training by updating them at each iteration. This allows the model to learn word properties that are significant for sentiment detection, such as the difference between antonyms, that are not necessarily captured well in the initial embeddings.

Focus: One crucial extension beyond the model in (Kim, 2014) is the *focus* vector, indicating the position(s) of a given company in the text. The focus vector is shown in darker grey in Figure 1, with the company position in a red frame. This provides an additional dimension to the word embedding, and helps to distinguish between training instances that differ only in focus and sentiment.

The reason for introducing focus is that sentiment is not a feature of the text as a whole, but of each company mention. Two mentions in the same text may have different sentiments and a model needs be able to distinguish them. In this sense, this task is similar to *aspect-based sentiment analysis* (Pontiki et al., 2016), where the task is not to classify a text or sentence, but an entity within the text. The notion of focus is similar to *attention* (Bahdanau et al., 2016; Yin et al., 2016), with the difference that attention is learned during training whereas focus is given as an additional input.

We experiment with three alternative representations for focus. The **baseline** model has no focus, and uses only lexical features without NEs. In the **binary** strategy, the focus vector contains ones in positions where the target company is appears, and zeros elsewhere. In the **smoothed** strategy, the focus value for each word indicates the *proximity* of the current word to the position of the nearest mention of the target company. Proximity is computed according to the formula:

$$Prox(p) = \frac{1}{1 + |p - m|}$$

where p is the position of the current word and m is the position of the nearest mention of the target company. Thus, proximity is 1 for a company mention, $1/2$ for its immediate neighbours, $1/3$ for the next neighbours, etc. It is never 0, which allows a convolution filter to use information about focus points, even if it exceeds the filter length.

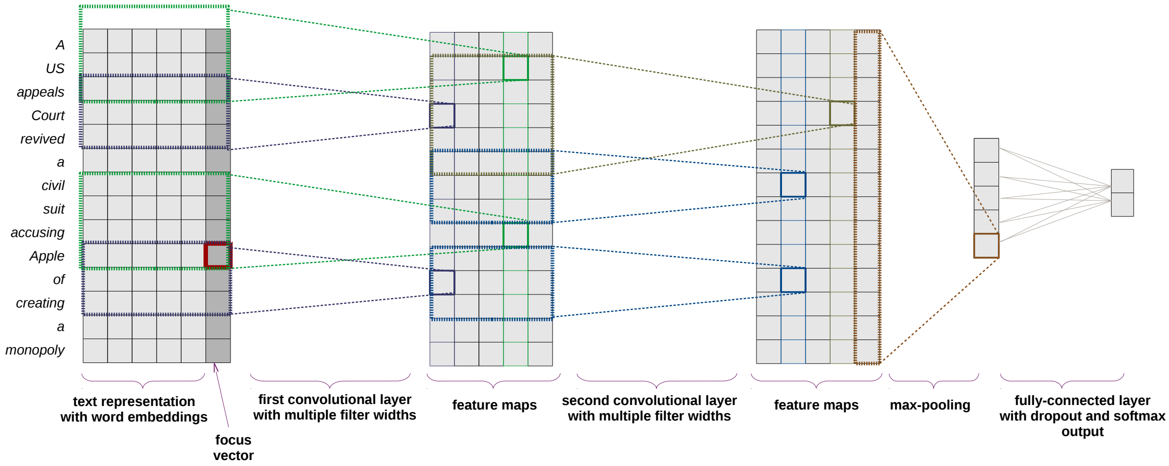
A
US
appeals
Court
revived
a
civil
suit
accusing
Apple
of
creating
a
monopoly
text representation with word embeddings
focus vector
first convolutional layer with multiple filter widths
feature maps
second convolutional layer with multiple filter widths
feature maps
max-pooling
fully-connected layer with dropout and softmax output

Figure 1: Model architecture with focus vector and two convolution layers

Data augmentation: Since the training set contains only "simple" instances—that mention exactly one company, as described in Section 2—we introduce a method for *data augmentation* which allows us to generate more realistic data. By feeding our model instances that mention *several* companies, we force the network to make use of the focus information, so it can learn to handle more complex test instances, producing a better model.

To augment the data we randomly select two simple instances—which gives them a 50% chance of having different sentiments—and concatenate them. We then randomly decide which of them should receive focus. As a result, we get an instance that mentions a focus company and a distractor company either on the left or on the right of the focus. We expect that using these examples the model would learn to ignore sentiment signals if they are far removed from the focus.

Model tuning: We have two different corpora—a large one collected by us and a small training set provided by the task organisers. We used a two-stage learning procedure, where the model is first trained using the large corpus and then it is refined using the shared task data. The core idea is that the first stage is used to learn a coarse solution for the problem on rich data, while the latter stage is used to fine-tune the model for the specific task at hand. In particular, in the second stage the model should calibrate an output to the exact values of the scores, since in the first stage all instances are labelled using only 1 or -1.

For the first training phase we used 10K sentences as a development set to determine when to stop training. For the second phase we take another approach since we want to use as much data as possible for training. First, we split the data into two halves and tune the model, using the second half as a development set to define the number of steps before it overfits. Then we tune the model using the entire training set (and no development set) and allow it to train the same number of *epochs*, which means the model has seen each training instance the same number of times.

4 Results

Table 1 shows the results for a selection of models trained on our data and tested on the shared task's *training set*. For the experiments we use only English microblogs. The evaluation is done in terms of *cosine* similarity between a model's output and the correct answer, as well as *accuracy*.[2]

We explore several hyper-parameters of the model: the number of convolution layers and the number and size of convolution filters. We also report the effect of using (or not) the data augmentation scheme described above. We also manipulate the instances, where the same company is mentioned several times, by considering instances with (**many**) foci or splitting them into several instances with only **one** focus point.

As shown in the table, the data augmentation scheme does not help the performance for this par-

[2] To compute accuracy, we map the sentiment score into three classes: negative (-1:-0.2), neutral (-0.2:0.2), and positive (0.2:1). This is a rather arbitrary split into three classes, which provides a rough estimate of the model's accuracy. In actual training we optimise the *loss*, i.e. the cross-entropy between the model's output and the true value.

Au	**Focus**	**FP**	**CL**	**Filter**	**Fi**	**Headlines**		**Blogs**	
						acc	**cos**	**acc**	**cos**
no	no (baseline)	one	1	3,4,5	128	**70.67**	**53.92**	63.59	32.71
no	binary	one	1	3,4,5	128	69.88	52.53	**63.98**	30.40
no	binary	many	1	3,4,5	128	70.05	52.15	60.14	26.59
no	smooth	one	1	3,4,5	128	68.48	51.36	58.91	25.91
no	smooth	one	2	3,4,5	128	70.32	53.35	62.03	**33.19**
yes	smooth	one	1	3,4,5	128	69.53	51.14	61.83	31.41
yes	smooth	many	1	3,4,5	128	70.32	52.11	58.91	21.52
yes	binary	one	3	3,4,5	128	70.40	52.78	62.09	30.43
yes	binary	one	1	3,4,5	200	68.91	49.77	61.31	27.91
yes	binary	many	1	3,4,5	128	70.05	50.09	61.64	29.63
yes	binary	many	3	3,4,5	128	69.00	50.28	60.60	26.03
yes	smooth	one	1	3,7,11	128	70.05	50.29	58.19	24.90
yes	smooth	one	2	3,7,11	128	69.53	49.53	63.00	29.13
yes	smooth	many	2	3,7,11	128	69.26	49.68	61.64	25.12
yes	binary	one	6	3,8	40	64.27	42.68	57.35	24.22

Table 1: A selection of best-performing models. Legend: **Au**—augmentation, **FP**—number of focus points per instance, **CL**—number of convolution layers, **Fi**—number of filters (of each size).

	Example	True score	Model output
1	*Tesco names Deloitte as new auditor after accounting scandal.*	-0.452	0.289
2	*Tesco breaks its downward slide by cutting sales decline in half.*	0.172	-0.703

Table 2: Problematic examples.

	Headlines	**Blogs**
without tuning	51.30	36.03
with tuning	67.95	60.73

Table 3: Official results for SemEval 2017 Task 5: cosine similarity.

ticular task. Thus, we submitted a solution without augmentation. Using foci increases performance for microblogs but not for headlines, probably because most instances in the task have only one mention. However, we submitted a solution with (smooth) focus since we believe it will be crucial in more realistic settings.

It can also be seen from the table that, although the results for headlines and microblogs have comparable accuracy, microblog classification is substantially worse in terms of cosine similarity.

The model we chose for the SemEval submission (for both subtasks) is highlighted in blue in the table. For each subtask, we made two submissions: one without tuning—using only our data, and one with the tuning step—we continue refining the model, using headlines and microblog data respectively. The final results of the shared task are shown in Table 3. As can be seen in the table, tuning provides a substantial improvement—16% for headlines and 24% for microblogs. Table 2 shows some examples of the more problematic cases that we found during error analysis.

Example 1 would require processing of long-distance dependencies. In this sentence the key phrase *accounting scandal* is far from the focus company *Tesco*, so none of the convolutional filters is applied to the company name and the phrase at the same time. The focus mechanism reduces the weight of the phrase, since another company name appears between the focus and the phrase, which may indicate a drawback of our model on such short input strings. Some sentences are incorrectly classified due to a complicated syntactic structure.

Example 2 contains a string of strongly negative cues (*breaks, downward slide, cutting, sales decline*), which should cancel each other out, but correct processing of such sentences would require deeper semantic analysis. Note, that in this task we have rather short pieces of text; in a more realistic setting the model should classify an entire document, where the company of interest would be mentioned multiple times with different keywords in context.

Acknowledgements

We are thankful to Denny Britz, whose TensorFlow implementation[3] of (Kim, 2014) model was used as a starting point for our implementation.

[3] https://github.com/dennybritz/cnn-text-classification-tf

References

Martin Atkinson, Jakub Piskorski, Erik van der Goot, and Roman Yangarber. 2011. Multilingual real-time event extraction for border security intelligence gathering. In U. Kock Wiil, editor, *Counterterrorism and Open Source Intelligence*, Springer Lecture Notes in Social Networks, Vol. 2, pages 355–390.

Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. 2016. End-to-end attention-based large vocabulary speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, pages 4945–4949.

Mian Du, Lidia Pivovarova, and Roman Yangarber. 2016. PULS: natural language processing for business intelligence. In *Proceedings of the 2016 Workshop on Human Language Technology*. Go to Print Publisher, pages 1–8.

Zafer Erenel and Hakan Altınçay. 2013. Improving the precision-recall trade-off in undersampling-based binary text categorization using unanimity rule. *Neural Computing and Applications* 22(1):83–100.

Siegfried Handschuh, Adamantios Koumpis, Ross McDermott, Keith Cortis, Laurentiu Vasiliu, and Brian Davis. 2016. Social sentiment indices powered by x-scores. In *2nd International Conference on Big Data, Small Data, Linked Data and Open Data, ALLDATA 2016*. INSIGHT Centre for Data Analytics, NUI Galway, Ireland.

Silja Huttunen, Arto Vihavainen, Mian Du, and Roman Yangarber. 2013. Predicting relevance of event extraction for the end user. In Thierry Poibeau, Horacio Saggion, Jakub Piskorski, and Roman Yangarber, editors, *Multi-source, Multilingual Information Extraction and Summarization*, Springer Berlin, Theory and Applications of Natural Language Processing, pages 163–176.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *EMNLP*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Lidia Pivovarova, Silja Huttunen, and Roman Yangarber. 2013. Event representation across genre. In *Proceedins of the* 1^{st} *Workshop on Events: Definition, Detection, Coreference, and Representation*. NAACL HLT.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphee De Clercq, Veronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Núria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. Association for Computational Linguistics, San Diego, California, pages 19–30. http://www.aclweb.org/anthology/S16-1002.

Efstathios Stamatatos. 2008. Author identification: Using text sampling to handle the class imbalance problem. *Information Processing & Management* 44(2):790–799.

Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*.

Roman Yangarber and Ralf Steinberger. 2009. Automatic epidemiological surveillance from on-line news in MedISys and PULS. In *Proceedings of IMED-2009: International Meeting on Emerging Diseases and Surveillance*. Vienna, Austria.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. ABCNN: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics* 4:259–272.