Neural Sentiment Analysis for a Real-World Application

Daniele Bonadiman[‡], **Giuseppe Castellucci**[†],
Andrea Favalli[†], **Raniero Romagnoli**[†], **Alessandro Moschitti**[‡◇]
[‡]Department of Computer Science and Information Engineering, University of Trento, Italy
[◇]Qatar Computing Research Institute, HBKU, Qatar
[†]Almawave Srl., Italy
d.bonadiman@unitn.it, amoschitti@gmail.com
{g.castellucci,a.favalli,r.romagnoli}@almawave.it

Abstract

English. In this paper, we describe our neural network models for a commercial application on sentiment analysis. Different from academic work, which is oriented towards complex networks for achieving a marginal improvement, real scenarios require flexible and efficient neural models. The possibility to use the same models on different domains and languages plays an important role in the selection of the most appropriate architecture. We found that a small modification of the state-of-the-art network according to academic benchmarks led to a flexible neural model that also preserves high accuracy.

Italiano. *In questo lavoro, descriviamo i nostri modelli di reti neurali per un'applicazione commerciale basata sul* sentiment analysis. *A differenza del mondo accademico, dove la ricerca è orientata verso reti anche complesse per il raggiungimento di un miglioramento marginale, gli scenari di utilizzo reali richiedono modelli neurali flessibili, efficienti e semplici. La possibilitá di utilizzare gli stessi modelli per domini e linguaggi variegati svolge un ruolo importante nella scelta dell'architettura. Abbiamo scoperto che una piccola modifica della rete allo stato dell'arte rispetto ai benchmarks accademici produce un modello neurale flessibile che preserva anche un'elevata precisione.*

1 Introduction

In recent years, Sentiment Analysis (SA) in Twitter has been widely studied. Its popularity has been fed by the remarkable interest of the industrial world on this topic as well as the relatively easy access to data, which, among other, allowed the academic world to promote evaluation campaigns, e.g., (Nakov et al., 2016), for different languages. Many models have been developed and tested on these benchmarks, e.g., (Li et al., 2010; Kiritchenko et al., 2014; Severyn and Moschitti, 2015; Castellucci et al., 2016). They all appear very appealing from an industrial perspective, as SA is strongly connected to many types of business through specific KPIs[1]. However, previous academic work has not provided clear indications on how to select the most appropriate learning architecture for industrial applications.

In this paper, we report on our experience on adopting academic models of SA to a commercial application. This is a social media and microblogging monitoring platform to analyze brand reputation, competition, the voice of the customer and customer experience. More in detail, sentiment analysis algorithms register customers' opinions and feedbacks on services and products, both direct and indirect.

An important aspect is that such clients push for easily adaptable and reliable solutions. Indeed, multi-tenant applications and sentiment analysis requirements cause a high variability of the approaches to the tasks within the same platform. This should be capable of managing multi-domain and multi-channel content in different languages as it provides services for several clients in different market segments. Moreover, scalability and lightweight use of computational resources preserving accuracy is also an important aspect. Finally, dealing with different client domains and data potentially requires constantly training new models with limited time availability.

To meet the above requirements we started from

[1]Key Performance Indicators are strategic factors enabling the performance measurement of a process or activity.

the state-of-the-art model proposed in (Severyn and Moschitti, 2015), which is a Convolutional Neural Network (CNN) with few layers mainly devoted to encoding a sentence representation. We modified it by adopting a recurrent pooling layer, which allows the network to learn longer dependencies in the input sentence. An additional benefit is that such simple architecture makes the network more robust to biases from the dataset, generalizing better on the less represented classes. Our experiments on the SemEval data in English as well as on a commercial dataset in Italian show a constant improvement of our networks over the state of the art.

In the following, Section 2 places the current work in the literature. Section 3 introduces the application scenario. Sections 4 and 5 presents respectively our proposal for a flexible architecture and the experimental results. Finally, Section 6 reports the conclusions.

2 Related Work

Although sentiment analysis has been around for one decade, a clear and exact comparison of models has been achieved thanks to the organization of international evaluation campaigns. The main campaign for SA in Twitter in English is SemEval, which has been organized since 2013. A similar campaign in the Italian language (SENTIPOLC) (Barbieri et al., 2016) is promoted within Evalita since 2014.

Among other approaches, Neural Networks (NNs), and in particular CNNs, outperformed the previous state of the art techniques (Severyn and Moschitti, 2015; Castellucci et al., 2016; Attardi et al., 2016; Deriu et al., 2016). Those systems share some architectural choices: (i) use of Convolutional Sentence encoders (Kim, 2014), (ii) leveraging pre-trained word2vec embeddings (Mikolov et al., 2013) and (iii) use of distant supervision to pre-train the network (Go et al., 2009). Despite this network is simple and provides state of the art results, it does not model long-term dependencies in the tweet by construction.

3 Application Scenario

Our commercial application is a social media and micro-blogging monitoring platform, which is used to analyze brand reputation, competitors, the voice of the customer and customer experience. It is capable of managing multi-domain and multi-channel content in different languages and it is provided as a service for several clients on different market segments.

The application uses an SA algorithm to analyze the customers' opinions and feedbacks on services and products, both direct and indirect. The sentiment metric is used by the application clients to point out customer experience, expectations, and perception. The final aim is to promptly react and identify improvement opportunities and, afterward, measure the impact of the adopted initiatives.

3.1 Focused Problem Description

Industrial applications, used by demanding clients, and dealing with real data tend to prefer easily adaptable and reliable solutions. Major problems are related to multi-tenant applications with several client requirements on the sentiment analysis problem, often requiring variations on task approaches within the same platform. Moreover, high attention is put on scalability and lightweight use of computational resources, preserving accurate performance. Finally, dealing with different client domains and data potentially requires constantly training new models with limited time availability.

3.2 Data Description

The commercial social media and micro-blogging monitoring platform continuously acquires data coming from several sources; among these, we selected Twitter data as the main source for our purposes.

First, the public Twitter stream was collected for several months without specific domain restriction to build the dataset used for the word embedding training. The total amount of tweets used accounts for 100 million Italian tweets and 50 million English tweets.

Then, a dataset has been constructed from a specific market sector in Italian. The data collection was performed on the public Twitter stream with specific word restriction performed in order to filter the tweets of interest on the automotive domain. Afterward, the commercial platform applies different techniques in order to exclude from these collections the tweets that are not relevant for the specific insight analysis.

The messages were then used to construct the dataset for our experiments. A manual annotation phase has been performed together with the

demanding client in order to best suit the insight objective requirement. Even though structured guidelines were agreed upon before creating the dataset and continuously checked against, this approach tended to generate dataset characteristics: in particular, unbalanced distribution of the examples over the different classes has been measured. It makes necessary a flexible model capable of handling such phenomena without the need of costly tuning phases and/or network reengineering.

4 Our Neural Network Approach

The task of SA in Twitter aims at classifying a tweet $t \in T$ into one of the three sentiment classes $c \in C$, where $C = \{positive, neutral, negative\}$. This can be achieved by learning function $f : T \rightarrow C$ through a neural network. The architecture here proposed is based on (Severyn and Moschitti, 2015) and it is structured in three steps: (i) a tweet is encoded into an embedding matrix, (ii) an encoder maps the tweet matrix into a fixed size vector and (iii) a single output layer (a logistic regression layer) classifies this vector over the three classes.

In contrast to Severyn and Moschitti (2015), we adopted a Recurrent Pooling layer that allows the network to learn longer dependencies in the input sentence (i.e. sentiment shifts). This architectural change makes the network less sensible to learn biases from the dataset and therefore generalize better on poorly represented classes.

Embedding: a tweet t is represented as a sequence of words $\{w_1, .., w_j, .., w_N\}$. Tweets are encoded into a sentence matrix $\mathbf{t} \in \mathbb{R}^{d \times |\mathbf{t}|}$, obtained by concatenating its word vectors $\mathbf{w}_j$, where d is the size of the word embeddings.

Sentence Encoder: it is a function that maps the sentence matrix $\mathbf{t}$ into a fixed size vector x representing the whole sentence. Severyn and Moschitti (2015) used a convolutional layer followed by a global max-pooling layer to encode tweets. The convolution operation applies a sliding window operation (with window of size m) over the input sentence matrix. More specifically, it applies a non-linear transformation generating an output matrix $\tilde{\mathbf{x}} \in \mathbb{R}^{N \times d_{conv}}$ where d_{conv} is the number of convolutional filters and N is the length of the sentence. The max-pooling operation applies an element-wise max operation to the transformed sentence matrix $\tilde{\mathbf{x}}$, resulting in a fixed size vector representing the whole sentence.

In this work, we propose to substitute the max-pooling operation with a Bidirectional Gated Recurrent Unit (BiGRU) (Chung et al., 2014; Schuster and Paliwal, 1997). The GRU is a Gated Recurrent Neural Network capturing long term dependencies over the input. A GRU processes the input in a direction (e.g., from left to right), updating a hidden state that keeps the memory of what the network has processed so far. In this way, a whole sentence can be represented by taking the hidden state at the last step. In order to capture dependencies in both directions, i.e., a stronger representation of the sentence, we apply a BiGRU, which performs a GRU operation in both the directions $BiGRU(\tilde{\mathbf{x}}) = [\overrightarrow{GRU}(\tilde{\mathbf{x}}); \overleftarrow{GRU}(\tilde{\mathbf{x}})]$.

Classification: the final module of the network is the output layer (a logistic regression) that performs a linear transformation over the sentence vector by mapping it in a d_{class} dimensional vector followed by a softmax activation, where d_{class} is the number of classes.

5 Experiments

5.1 Setup

Similarly to Severyn and Moschitti (2015), for the CNN, we use a convolutional operation of size 5 and $d_{conv} = 128$ with rectified linear unit activation, ReLU. For the BiGRU, we use 150 hidden units for both $\overleftarrow{GRU}$ and $\overrightarrow{GRU}$ obtaining a fixed size vector of size 300.

Word embeddings: for all the proposed models, we pre-initialize the word embedding matrices with the standard skip-gram embedding of dimensionality 50 trained on tweets retrieved from the Twitter Stream.

Training: the network is trained using SGD with shuffled mini-batches using the Adam update rule (Kingma and Ba, 2014) and an early stopping (Prechelt, 1998) strategy with patience $p = 10$. Early stopping allows avoiding overfitting and to improve the generalization capabilities of the network. Then, we opted for adding dropout (Srivastava et al., 2014) with rates of 0.2 to improve generalization and avoid co-adaptation of features (Srivastava et al., 2014).

Datasets: we trained and evaluated our architecture on two datasets: the English dataset of Semeval 2015 (Rosenthal et al., 2015) described by

Table 1: Splits of the Semeval dataset

	pos.	neu.	neg.	total
train	5,895	471	3,131	9,497
valid	648	57	430	1,135
test 2013	2,734	160	1,541	4,435
test 2015	1,899	190	1,008	3,097

Table 2: Splits of the Italian dataset

	pos	neu	neg	total
train	4,234	6,434	2,170	12,838
valid	386	580	461	1,427
test	185	232	83	500

Table 3: English results on the SemEval dataset

	2013 test		2015 test	
	$F1$	$F1_{p,n}$	$F1$	$F1_{p,n}$
S&M (2015)	—	72.79	—	64.59
CNN+Max	**72.04**	67.71	67.14	62.63
CNN+BiGRU	71.67	**68.10**	**68.03**	**63.82**

Table 4: Italian results on the automotive dataset

	Valid		Test	
	$F1$	$F1_{p,n}$	$F1$	$F1_{p,n}$
CNN+Max	**65.34**	62.35	**69.35**	62.88
CNN+BiGRU	64.85	**67.71**	68.32	**67.55**

Table 1 in terms of the size of the data splits and positive, negative and neutral instances. We used the validation set for parameter tuning and to apply early stopping whereas the systems are evaluated on the two test sets of 2013 and 2015, respectively.

The Italian dataset was built in-house for the automotive domain: we collected from the Twitter stream as explained in Section 3.2 and divided it into three different splits for training, validation and testing, respectively. Table 2 shows the size of the splits. Due to the nature of the domain, many tweets in the dataset are neutral or objective, this makes the label distribution much different from the usual benchmarks. For example, the neutral class is the least represented in the English dataset (see Table 1) and the most represented in the Italian data. The imbalance can potentially bias neural networks towards the most represented class. One of the features our approach is to diminish such effect.

Evaluation metrics: we used the following evaluation metrics, Macro-F1 (the average of the F1 over the three sentiment categories). Additionally, we report the $F1_{p,n}$, which is the average F1 of the positive and negative class. This metric is the official evaluation score of the SemEval competition.

5.2 Results on English Data

Table 3 presents the results on the English dataset of SemEval 2015. The first row shows the outcome reported by Severyn and Moschitti (2015) (S&M). CNN+Max is a reimplementation of the above system with Convolution and Max-Pooling but trained just on the official training data without distant supervision. This system is used as a strong baseline in all our experiments. Lastly, we report the results obtained with the BiGRU pooling strategy described in Section 4. The proposed architecture presents a slight improvement over the strong baseline (~ 1 point of both $F1$ and $F1_{p,n}$ score on the test).

5.3 Results on Italian Data

Table 4 presents the result on the Italian dataset. Despite that on this dataset the proposed CNN+BiGRU model obtains lower F1 scores, it shows improved performance in terms of $F1_{p,n}$ (5 points on both validation and test sets). This suggests that the proposed model tends to generalize better on the less represented classes, which, in the case of the Italian training dataset, are the positive and negative classes (as pointed out in Table 2).

5.4 Discussion of the Results

We analyzed the classification scores of some words to show that our approach is less affected by the skewed distribution of the dataset. The sentiment trends, as captured by the neural network in terms of scores, are shown in Table 5.4). For example, the word *Mexico* classified by CNN+Max produces the scores, 0.06, 0.35, 0.57, while CNN+BiGRU outcome, 0.18, 0.52, 0.30, for the negative, neutral and positive classes, respectively. This shows that CNN+BiGRU is less biased by the data distribution of the sampled word in the dataset, which is, 0, 1, 5, i.e., *Mexico* appears 5 times more in positive than in neutral messages and never in negative messages.

This skewed distribution biased more CNN+Max as the positive class gets 0.57 while the negative one only 0.06. CNN+BiGRU is able, instead, to recover the correct neutral class. We believe that CNN+Max is more influenced by

	Cnn+Max	Cnn+BiGRU
Mexico	(.06, .35, .57)	(.18, .51, .30)
Italy	(.06, .54, .38)	(.18, .54, .26)
nice	(.007, .009, .98)	(.05, .07, .87)

Table 5: Word classification scores obtained with the two neural architectures on English language. The scores refer to the negative, neutral and positive classes, respectively.

the distribution bias as the max pooling operation seems to capture very local phenomena. In contrast, BiGRU exploits the entire word sequence and thus can better capture larger informative context.

A similar analysis in Italian shows the same trends. For example, the word *panda* is classified as, 0.05, 0.28, 0.66, by CNN+Max and 0.07, 0.56, 0.35 by CNN+BiGRU, for negative, neutral and positive classes, respectively. Again, the distribution in the Italian training set of this word is very skewed towards the positive class: it confirms that CNN+Max is more influenced by the distribution bias, while our architecture can better deal with it.

6 Conclusions

In this paper, we have studied state-of-the-art neural networks for the Sentiment Analysis of Twitter text associated with a real application scenario. We modified the network architecture by applying a recurrent pooling layer enabling the learning of longer dependencies between words in tweets. The recurrent pooling layer makes the network more robust to unbalanced data distribution. We have tested our models on the academic benchmark and most importantly on our data derived from a real-world commercial application. The results show that our approach works well for both English and Italian languages. Finally, we observed that our network suffers less from the dataset distribution bias.

References

Giuseppe Attardi, Daniele Sartiano, Chiara Alzetta, and Federica Semplici. 2016. Convolutional neural networks for sentiment analysis on italian tweets. In *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016), Napoli, Italy, December 5-7, 2016*. CEUR-WS.org.

Francesco Barbieri, Valerio Basile, Danilo Croce, Malvina Nissim, Nicole Novielli, and Viviana Patti. 2016. Overview of the evalita 2016 sentiment polarity classification task. In *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*.

Giuseppe Castellucci, Danilo Croce, and Roberto Basili. 2016. Context-aware convolutional neural networks for twitter sentiment analysis in italian. In *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016), Napoli, Italy, December 5-7, 2016*. CEUR-WS.org.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Jan Deriu, Maurice Gonzenbach, Fatih Uzdilli, Aurelien Lucchi, Valeria De Luca, and Martin Jaggi. 2016. Swisscheese at semeval-2016 task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision. In *SemEval@ NAACL-HLT*, pages 1124–1128.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter sentiment classification using distant supervision.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL.

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

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M. Mohammad. 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50(1):723–762, May.

Shoushan Li, Sophia Yat Mei Lee, Ying Chen, Chu-Ren Huang, and Guodong Zhou. 2010. Sentiment classification and polarity shifting. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 635–643. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. Semeval-2016 task 4: Sentiment analysis in twitter. In *SemEval@ NAACL-HLT*, pages 1–18.

Lutz Prechelt. 1998. Early stopping-but when? In *Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop*, pages 55–69, London, UK, UK. Springer-Verlag.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif Mohammad, Alan Ritter, and Veselin Stoyanov. 2015. Semeval-2015 task 10: Sentiment analysis in twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 451–463, Denver, Colorado, June. Association for Computational Linguistics.

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

Aliaksei Severyn and Alessandro Moschitti. 2015. Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 959–962. ACM.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.