Aspect-based Sentiment Analysis: X2Check at ABSITA 2018

Emanuele Di Rosa
Chief Technology Officer
App2Check s.r.l.
emanuele.dirosa
@app2check.com

Alberto Durante
Research Scientist
App2Check s.r.l.
alberto.durante
@app2check.com

Abstract

English. In this paper we describe and present the results of the two systems, called here X2C-A and X2C-B, that we specifically developed and submitted for our participation to ABSITA 2018, for the Aspect Category Detection (ACD) and Aspect Category Polarity (ACP) tasks. The results show that X2C-A is top ranker in the official results of the ACD task, at a distance of just 0.0073 from the best system; moreover, its post deadline improved version, called X2C-A-s, scores first in the official ACD results. About the ACP results, our X2C-A-s system, which takes advantage of our ready-to-use industrial Sentiment API, scores at a distance of just 0.0577 from the best system, even though it has not been specifically trained on the training set of the evaluation.

Italiano. *In questo articolo descriviamo e presentiamo i risultati dei due sistemi, chiamati qui X2C-A e X2C-B, che abbiamo specificatamente sviluppato per partecipare ad ABSITA 2018, per i task Aspect Category Detection (ACD) e Aspect Category Polarity (ACP). I risultati mostrano che X2C-A si posiziona ad una distanza di soli 0.0073 dal miglior sistema del task ACD; inoltre, la sua versione migliorata, chiamata X2C-A-s, realizzata successivamente alla scadenza, mostra un punteggio che lo posiziona al primo posto nella classifica ufficiale del task ACD. Riguardo al task ACP, il sistema X2C-A-s che utilizza il nostro standard Sentiment API, consente di ottenere un punteggio che dista solo 0.0577 dal miglior sistema, nonostante il classificatore di sentiment non sia stato specificamente addestrato sul training set della evaluation.*

1 Introduction

The traditional task of sentiment analysis is the classification of a sentence according to the positive, negative, or neutral classes. However, such task in this simple version is not enough to detect when a sentence contains a mixed sentiment, in which a positive sentiment is referred to one aspect and a negative sentiment to another aspect. Aspect-based sentiment analysis is focused on the sentiment classification (negative, neutral, positive) for a given aspect/category in a sentence. In nowadays world, reviews became an important tool widely used by consumers to evaluate services and products. Given the large amount of reviews available online, systems allowing to automatically classify reviews according to different categories, and assign a sentiment to each of those categories, are gaining more and more interest in the market. The former task is called Aspect Category Detection (ACD) since detects whether a review speaks about one of the categories under evaluation; the latter task, called Aspect Category Polarity (ACP) tries to assign a sentiment independently for each aspect. In this paper, we present X2C-A and X2C-B, two different implementations for dealing with the ACD and ACP tasks, specifically developed for the ABSITA evaluation (Basile et al., 2018). In particular, we describe the models used to participate to the ACD competition together with some post deadline results, in which we had the opportunity to improve our ACD results and evaluate our systems also on the ACP task. The resuls show that our X2C-A system is top ranking in the official ACD competition and scores first, in its X2C-A-s version. Moreover, by testing our ACD models on the ACP tasks, with the help of our standard X2Check sentiment API, the X2C-A-s system scores fifth at a

distance of just 0.057 from the best system, even if the other systems have a sentiment classifier specifically trained on the training set of the competition. This paper is structured as follow: after the introduction we present the descriptions of our two systems submitted to ABSITA and the results on the development set; then we show and discuss the results on the official testset of the competition for both ACD and ACP, finally we provide our conclusions.

2 Systems description

The official training dataset has been split into our internal training set (80% of the documents) and development set (the remaining 20%). We randomly sampled the examples for each category, thus obtaining different sets for training/test set, by keeping the per category distribution of the samples through the three sets. We submitted two runs, as the results of the two different systems we developed for each category, called X2C-A and X2C-B. The former has been developed on top of the Scikit-learn library in Python language (Pedregosa et al., 2011), and the latter on top of the WEKA library (Frank et al., 2016) in JAVA language. In both cases, the input text has been cleaned with a typical NLP pipeline, involving punctuation, numbers and stopwords removal. The two systems have been developed separately, but the best algorithms obtained by both the model selections are different implementations of Support Vector Machine. More details in the following sections.

2.1 X2C-A

The X2C-A system has been created by applying an NLP pipeline including a vectorization of the collection of reviews to a matrix of token counts of the bi-grams; then, the count matrix has been transformed to a normalized tf-idf representation (term-frequency times inverse document-frequency). As machine learning algorithm, an implementation of the Support Vector Machine has been used, specifically the LinearSVC. Such algorithm has been selected as the best performer on such dataset compared to other common implementations available in the sklearn library.

Table 1 shows the F1 score on the positive label in the development set for each category, where the average value on all of the categories is 84.92%. X2C-A shows the lowest performance on the Value category, while shows the best performance on Location, and high score on Wifi and Staff.

2.2 X2C-B

In the model selection process, the two best algorithms have been Naive Bayes and SMO. We built a model with both algorithms for each category. We took into account the F1 score on the positive labels in order to select the best algorithm. In this implementation, SMO (Sequential Minimal Optimization) (Platt, 1998) (Keerthi et al., 2001) (Hastie et al., 1998) has been the best performing algorithm on all of the categories, and showed an average F1 score across all categories of 85.08%. Its scores are reported in Table 1, where we also compare its performance with the X2C-A one on the development set.

The two systems are built on different implementation of support vector machines, as previously pointed out, and differ on the features extraction process. In fact, X2C-B takes into account a vocabulary of the 1000 most mentioned words in the training set, according to the size limit parameter available in the StringToWordVector Weka function. Moreover, it uses unigrams instead of the bi-grams extraction performed in X2C-A. The two systems reach similar results, i.e. high scores on Location, Wifi and Staff, and low scores on the Value category. However, the overall weighted performance is very close, around 85% of F1 on the positive labels, and since for some categories is better X2C-A and for others X2C-B, we decided to submit both implementations, in order to understand which is the best one on the test set of the ABSITA evaluation.

Category	X2C-A	X2C-B
Cleanliness	0.8675	**0.8882**
Comfort	**0.8017**	0.7995
Amenities	**0.8041**	0.7896
Staff	0.8917	**0.8978**
Value	**0.7561**	0.7333
Wifi	0.9056	**0.9412**
Location	**0.9179**	0.9058

Table 1: F1 score per category on the positive labels on the development set. Best system in bold.

3 Results on the ABSITA testset

3.1 Aspect Category Detection

Table 2 shows the official results of the Aspect-based Category Detection task, with the addition of two post deadline results obtained by an additional version of X2C-A and X2C-B, called X2C-A-s and X2C-B-s.

The difference between the submitted results and the versions called X2C-A-s and X2C-B-s, is just at prediction time: X2C-A and X2C-B make a prediction at document-level, i.e. on the whole review, while X2C-A-s and X2C-B-s make a prediction at sentence-level, where each sentence has been obtained by splitting the reviews on some punctuation and key conjunction words. This makes more likely that each sentence contains one category and it seems to be easier for the models the category detection. For example, the review

> *The sight is beautiful, but the staff is rude*

is about Location and Staff, but since only a part of it is about Location, the location model of this category would receive a document containing "noise" from its point of view. In the post deadline runs, we reduce the "noise" by splitting this example review in *The sight is beautiful* which is only about Location, and *but the staff is rude* which is only about Staff. As we can see in Table 2, the performance of X2C-A increased significantly and reached a performance score that is better even than the first classified. However, the performance of X2C-B slighted decreased in its X2C-B-s version. This means that the model of this latter system is not helped by this kind of "noise" removal technique. This last result shows that such approach does not have a general applicability but it depends on the model; however, it shows to work very well on X2C-A.

In order to identify the categories where we perform better, we calculated the score of our systems on each category[1], as shown in Table 3 and Table 4. In Table 3 X2C-A is the best of our systems on all the categories except Cleanliness and Wifi, where X2C-B has reached the higher score. In Table 4, X2C-A-s shows the best performance on all of the categories. By comparing the results across tables 3 and 4, we can see that X2C-A-s is the best system on all of the categories, with the exception of Cleanliness, where X2C-B shows a slightly better performance. Comparing the results on development set (Table 1) and the ones on the ABSITA test set, Value is confirmed being the most difficult category to detect for our systems, with a score of 0.6168. Instead, concerning Wifi, which has been the easiest category in Table 1, in Table 4 shows a lower relative score, while the easiest category to detect overall was Location, on which X2C-A-split has reached a score of 0.8898.

[1]To obtain these scores, we modified the ABSITA evaluation script so that only one category is taken into account.

Team	Mic-Pr	Mic-Re	Mic-F1
X2C-A-s	0.8278	**0.8014**	**0.8144**
1	0.8397	0.7837	0.8108
2	0.8713	0.7504	0.8063
3	0.8697	0.7481	0.8043
X2C-A	0.8626	0.7519	0.8035
5	0.8819	0.7378	0.8035
X2C-B	**0.8980**	0.6937	0.7827
X2C-B-s	0.8954	0.6855	0.7765
7	0.8658	0.697	0.7723
8	0.7902	0.7181	0.7524
9	0.6232	0.6093	0.6162
10	0.6164	0.6134	0.6149
11	0.5443	0.5418	0.5431
12	0.6213	0.433	0.5104
baseline	0.4111	0.2866	0.3377

Table 2: ACD results

	X2C-A	X2C-B
Cleanliness	0.8357	**0.8459**
Comfort	**0.794**	0.7475
Amenities	**0.8156**	0.7934
Staff	**0.8751**	0.8681
Value	**0.6146**	0.6141
Wifi	0.8403	**0.8667**
Location	**0.8887**	0.8681

Table 3: X2Check per category results submitted to ACD

3.2 Aspect Category Polarity

In Table 5 we show the results of the Aspect-based Category Polarity task to which X2Check did not formally participate. In fact, after the evaluation deadline we had time to work on the ACP task.

In order to deal with the ACP task, we decided to take advantage of our ready-to-use, standard

	X2C-A-s	X2C-B-s
Cleanliness	**0.8445**	0.8411
Comfort	**0.8099**	0.739
Amenities	**0.8308**	0.7884
Staff	**0.8833**	0.8652
Value	**0.6168**	0.6089
Wifi	**0.8702**	0.8667
Location	**0.8898**	0.8584

Table 4: X2Check per category results submitted post deadline to ACD.

X2Check sentiment API (Di Rosa and Durante, 2017). In fact, since we do have an industrial perspective, we realized that in a real world setting, the fact of training an Aspect-based sentiment system through a specific training set has a high effort associated and cannot have a general purpose application. In fact, a very common case is the one in which new categories to predict have to be quickly added into the system. In this setting, a high effort activity of labeling examples for the training set would be required. Moreover, labeling a review according to the aspects mentioned and additionally assign a sentiment to each aspect requires a higher human effort than just labeling the category. For this reason, we decided to not specifically train a sentiment predictor specialized on the given categories/aspects in the evaluation. Thus, we performed an experimental evaluation in which after the prediction of the category in the review, our standard X2Check sentiment API has been called to predict the sentiment. Since we are aware that a review may, in general, speak about multiple aspects and having different sentiment associated, we decided to apply the X2C-A-s and X2C-B-s versions which use the splitting method described in section 3.1. More specifically:

1. each review document has been split into sentences

2. both the X2Check sentiment API and the X2C-A/X2C-B category classifiers were run on each sentence. The former gives as output the polarity of each sentence; our assumption is that each portion of the review has a high probability to have just one sentiment associated. The latter gives as output all of the detected categories in each sentence

3. the overall result of a review is given by the collection of all of the category-sentiment pairs found in the sentences

The results shown in Table 5 show that our assumption is valid. In fact, despite being a single sentiment model for all of the categories, we reach the fifth place in the official ranking with our X2C-A-s system, at a distance of just 0.057 from the best system specifically trained on such training set. Furthermore, the ACP performance depends on the ACD results, in fact the former task cannot reach a performance higher than the other. For this reason, we decided to evaluate the sentiment performance reached on the reviews whose categories have been correctly predicted. Thus, we created a score capturing the relationship between the two results: it is the ratio between the micro F1 score obtained in the ACP task and the one obtained in the ACD task. This hand crafted score shows the quality of the sentiment model, by removing the influence of the performance on the ACD task. The overall sentiment score obtained is 88.0% for X2C-B and 87.1% for X2C-A, showing that even if a specific train has not been made, the general purpose X2Check sentiment API shows very good results (recall that, according to (Wilson et al., 2009) humans agree in the sentiment classification in the 82% of cases).

Team	**Mic-Pr**	**Mic-Re**	**Mic-F1**
1	0.8264	0.7161	**0.7673**
2	0.8612	0.6562	0.7449
3	0.7472	0.7186	0.7326
4	0.7387	**0.7206**	0.7295
X2C-A-s	0.7175	0.7019	0.7096
5	**0.8735**	0.5649	0.6861
X2C-B-s	0.7888	0.6025	0.6832
6	0.6869	0.5409	0.6052
7	0.4123	0.3125	0.3555
8	0.5452	0.2511	0.3439
baseline	0.2451	0.1681	0.1994

Table 5: ACP results.

Tables 6 and 7 show for each category the micro-F1 and the sentiment score of the ACP task, calculated like in Table 4, and the relationship between ACP and ACD scores per category. We can see that the sentiment model has reached a very good performance on Cleanliness, Comfort, Staff and Location since it is close or over the 90%. However, like noticed for the ACD results, it is

difficult to handle reviews about the Value category.

	Micro-F1	SS
Cleanliness	0.7739	91.6%
Comfort	0.7165	88.5%
Amenities	0.6618	79.7%
Staff	0.8086	91.5%
Value	0.4533	73.5%
Wifi	0.6615	76.0%
Location	**0.8168**	**91.8%**

Table 6: X2C-A ACP results and sentiment score by category.

	Micro-F1	SS
Cleanliness	0.7626	90.7%
Comfort	0.671	90.8%
Amenities	0.6276	79.6%
Staff	0.7948	91.9%
Value	0.4581	75.2%
Wifi	0.6441	74.3%
Location	**0.7969**	**92.8%**

Table 7: X2C-B ACP results and sentiment score by category.

4 Conclusions

In this paper we presented a description of two different implementations for dealing with the ACD and ACP tasks at ABSITA 2018. In particular, we described the models used to participate to the ACD competition together with some post deadline results, in which we had the opportunity to improve our ACD results and evaluate our systems also on the ACP task. The resuls show that our X2C-A system is top ranking in the official ACD competition and scores first, in its X2C-A-s version. Moreover, by testing our ACD models on the ACP tasks, with the help of our standard X2Check sentiment API, the X2C-A-s system scores fifth at a distance of just 0.057 from the best system, even if the other systems have a sentiment classifier specifically trained on the training set of the competition.

References

Pierpaolo Basile, Valerio Basile, Danilo Croce and Marco Polignano. 2018. *Overview of the EVALITA 2018 Aspect-based Sentiment Analysis task (ABSITA)* in Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso, editors, Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18), CEUR.org, Turin

Eibe Frank, Mark A. Hall, and Ian H. Witten. 2016. The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E. 2011. *Scikit-learn: Machine Learning in Python* in Journal of Machine Learning Research, pp. 2825–2830.

Emanuele Di Rosa and Alberto Durante. LREC 2016 *App2Check: a Machine Learning-based system for Sentiment Analysis of App Reviews in Italian Language* in Proc. of the 2nd International Workshop on Social Media World Sensors, pp. 8-11.

Emanuele Di Rosa and Alberto Durante. 2017. *Evaluating Industrial and Research Sentiment Analysis Engines on Multiple Sources* in Proc. of AI*IA 2017 Advances in Artificial Intelligence - International Conference of the Italian Association for Artificial Intelligence, Bari, Italy, November 14-17, 2017, pp. 141-155.

Sophie de Kok, Linda Punt, Rosita van den Puttelaar, Karoliina Ranta, Kim Schouten and Flavius Frasincar. 2018. *Review-aggregated aspect-based sentiment analysis with ontology features* in Prog Artif Intell (2018) 7: 295.

Theresa Wilson, Janyce Wiebe and Paul Hoffmann. 2009. *Recognizing Contextual Polarity: An Exploration of Features for Phrase-Level Sentiment Analysis* in Computational Linguistic, pp. 399–433.

Seth Grimes. 2010. *Expert Analysis: Is Sentiment Analysis an 80% Solution?* http://www.informationweek.com/software/information-management/expert-analysis-is-sentiment-analysis-an-80–solution/d/d-id/1087919

J. Platt. 1998. *Fast Training of Support Vector Machines using Sequential Minimal Optimization* in Advances in Kernel Methods - Support Vector Learning

S.S. Keerthi and S.K. Shevade and C. Bhattacharyya and K.R.K. Murthy. 2001. *Improvements to Platt's SMO Algorithm for SVM Classifier Design* in Neural Computation, volume 13, pp. 637-649.

Trevor Hastie and Robert Tibshirani 1998. *Classification by Pairwise Coupling* in Advances in Neural Information Processing Systems, volume 10.