DFKI-DKT at SemEval-2017 Task 8: Rumour Detection and Classification using Cascading Heuristics

Ankit Kumar Srivastava
DFKI GmbH
Alt-Moabit 91c, 10559 Berlin, DE
ankit.srivastava@dfki.de

Georg Rehm
DFKI GmbH
Alt-Moabit 91c, 10559 Berlin, DE
georg.rehm@dfki.de

Julian Moreno Schneider
DFKI GmbH
Alt-Moabit 91c, 10559 Berlin, DE
julian.moreno_schneider@dfki.de

Abstract

We describe our submissions for SemEval-2017 Task 8, Determining Rumour Veracity and Support for Rumours. The Digital Curation Technologies (DKT) (Rehm and Sasaki, 2016, 2015) team at the German Research Center for Artificial Intelligence (DFKI) participated in two subtasks: Subtask A (determining the stance of a message) and Subtask B (determining veracity of a message, closed variant). In both cases, our implementation consisted of a Multivariate Logistic Regression (Maximum Entropy) classifier coupled with hand-written patterns and rules (heuristics) applied in a post-process cascading fashion. We provide a detailed analysis of the system performance and report on variants of our systems that were not part of the official submission.

1 Introduction

In today's digital age, the social, political and economic relevance of online media and online content is becoming more and more relevant. Accordingly, the task of analysing and determining the veracity of online content is receiving a growing amount of attention by the NLP community. The ability to detect whether a piece of news is fake or not, and to do so automatically, is a very timely language technology application (Zubiaga and Ji, 2014). Through these shared tasks, we intend to address which linguistic and contextual features characterise a rumour.

SemEval2017 Task 8 (Derczynski et al., 2017) provided all participants with a dataset consisting of tweets in response to breaking news stories. It contains conversations responding to rumourous tweets. These tweets have been annotated for support, deny, query or comment (SDQC). The competition consisted of two subtasks:

- **Subtask A:** Determining whether response tweets support, deny, query or comment (SDQC) on rumours (source tweet)
- **Subtask B:** Given a tweet, determine whether the statement is true or false (i. e., a rumour). This subtask featured two variants: closed (determining veracity from the tweet alone) and open (determining veracity from additional context). We participated in the closed task.

Our approach to both subtasks involved extracting relevant features from the provided data and training a classifier followed by a set of heuristics implemented in a cascading decision tree style (Minguillon, 2002). These rules, applied as a post-process, help induce a better mapping from classification results to rumour categorisation and veracity detection because they take into account specific features characterising a particular class.

In this paper we seek to answer two questions using Rumour Detection and Classification as a case-study:

- Which features comprise the set of post-process rules?
- What is the optimal technique to implement these heuristics (cascading order)?

This paper is structured as follows. Section 2 gives a bird's eye overview of our systems submitted for evaluation. Section 3 describes the various rumour detection and classification models as well as experimental setups (not part of the official submission). Section 4 displays the results and analyses them. Section 5 contains a discussion of the task in general followed by an explanation of some design decisions.

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 486–490,
Vancouver, Canada, August 3 - 4, 2017.

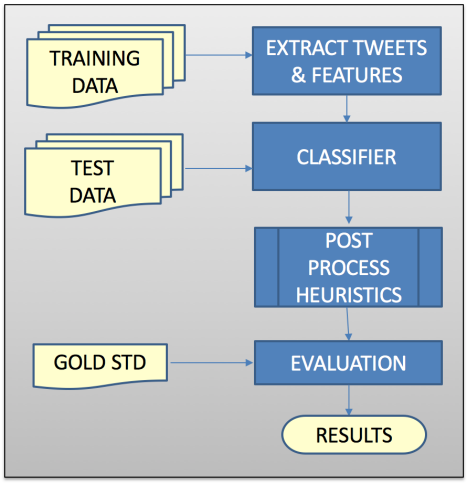

Figure 1: Workflow of the DFKI-DKT System for both tasks

Category	Subtask A	Subtask B
Training	4238 tweets	272 tweets
Development	281 tweets	25 tweets
Test	1049 tweets	28 tweets

Table 1: Overview of Training and Testing data

2 DFKI-DKT's Submission Overview

Our submissions can be categorised as hybrid systems since they consist of both machine learning and rule-based (heuristics) modules.

The first step was to extract contextual features (tweet text) and metadata features (Twitter user account properties and message properties) from the provided test data. We then trained a Maximum Entropy classifier (Malouf, 2002) followed by a set of heuristics (if-then clauses) implemented in a cascading decision tree style (Minguillon, 2002), see Figure 1.

2.1 Data and Tools

In terms of tools and resources we did not use any external data. All models were trained on the provided twitter dataset. Table 1 gives an overview of the size of the data for subtasks A and B. We implemented feature vector-based text classification models using the Mallet Machine Learning Toolkit (McCallum, 2002) in Java. The heuristics were implemented in the form of an experimentally determined sequence of if-then decision rules written in Python. Evaluation was performed using the scoring scripts provided by the task organisers.

2.2 Preprocessing

We employed the standard tokenisation scripts while extracting the feature vectors for training a classifier. We did not implement any other preprocessing step. In fact, it was discovered that cleaning the tweets actually impacted the classification algorithm in a negative way. We believe that certain as-is characteristics of the text (uppercase, spelling errors, emoticons, etc.) help in better distinguishing the used categories (SDQC).

2.3 Subtask A Heuristics

The classifier was trained on four classes (SDQC). This was followed by a post-processing module of decision rules based on linguistic patterns and Twitter metadata. The heuristics were as follows:

- **If** a tweet begins with a wh-word (where, when, how, what, why, which) and/or ends with a question mark, **then** classify it as *query*
- **If** a tweet contains a negation, **then** classify it as *denial*
- **If** a tweet is a retweet, **then** classify it as *support*
- **If** more than 70% of the text is all uppercase, **then** classify it as *comment*

2.4 Subtask B (closed) Heuristics

The classifier was trained on two classes (true, false). This was followed by a post-processing module of decision rules based on linguistic patterns and Twitter metadata. The heuristics were as follows:

- **If** a tweet begins with a wh-word (where, when, how, what, why, which) and/or ends with a question mark, **then** classify it as *false*
- **If** the tweet has been retweeted *x* number of times, **then** classify it as *true*
- **If** more than 70% of the text is all uppercase, **then** classify it as *false*
- **If** the tweet contains more than three @usernames and hashtags, **then** classify it as *false*
- **If** the author of the tweet as more than 10000 followers, **then** classify it as *true*

Pattern	Support	Query	Deny	Comment
RT (retweet)	**7.5%**	2.6%	6.2%	5.7%
@username (replies)	64.7%	**95.9%**	88.9%	92.1%
! (exclamation mark)	7.5%	6.7%	11.1%	**12.2%**
Negative emoticons	0.2%	0.3%	0.3%	**1.0%**
Positive emoticons	0%	**0.9%**	0.5%	0.2%
? (question mark)	6.7%	**65.8%**	14.9%	10.3%
Wh-word	6.5%	**21.3%**	13.4%	10.3%
"" (quotation marks)	5.2%	2.3%	**6.5%**	4.8%
Abusive language	2.5%	2.0%	**12.9%**	9.4%

Table 2: Percentage of tweets in the four categories of training data containing a specific feature.

3 Models and Experiments

In this section, we describe the details of the features used in our models as well as the different experimental settings.

3.1 Models

We trained three different classifiers, followed by applying the heuristics model described in Sections 2.3 and 2.4:

- Maximum Entropy classification (MaxEnt) (Malouf, 2002), also known as Multivariate Logistic Regression.
- Naive Bayes classification (Frank and Bouckaert, 2006) assumes independence of the features while counting.
- Winnow classification (Winnow2) (Littlestone, 1988) is similar to the perceptron model but uses a multiplicative weight update scheme rather than an additive method.

While we submitted only the MaxEnt model due to time constraints, we also include the results and analysis of the performance of the Naive Bayes and Winnow classifiers. We also computed an ensemble classifier, i. e., a voting-based combination of the three models' results using the following algorithm:

- Count the number of votes (MaxEnt, Naive Bayes, Winnow) for each of the categories (four for Subtask A, two for Subtask B)
- Select the category with the maximum number of votes
- If there is a tie, select the result of MaxEnt classifier

3.2 Useful Features

For subtask A (determining the category of a message), we compiled a list of distinctive features[1] characteristic of each of the stances: support, query, deny, comment. We conducted an investigation into linguistic and context-specific patterns that may distinguish one stance from the other. For example, query messages almost always have a wh-word and a question mark.

1. Message is a retweet, i. e., begins with *RT*
2. Message is a reply (*@usernames*)
3. Message contains exclamation marks
4. Message is a question (question mark or wh-word: who/what/when/why/where/how)
5. Message contains emoticons (smileys)
6. At least 70% of the message is in uppercase
7. Message contains negations (not, doesn't)
8. Message contains expletives or abuse

Table2 gives a snapshot of the frequency of the patterns on the training data in each of the SQDC categories.

3.3 Experimental Setup

The features used in the classification algorithms consisted of a vector of the words (twitter text). When we attempted to incorporate some of the features described above in the classification algorithm, the performance deteriorated. This led us to implement a post-process heuristic module and subject the results of the classification to a second

[1] After conducting a statistical analysis of the training data, we also used some of these features in determining the rumour veracity in subtask B, see Section 2.4.

System	Subtask A	Subtask B
MaxEnt+Heuristics	0.635	0.393
NaiveBayes+Heuristics	0.621	0.387
Winnow+Heuristics	0.630	0.400
Ensemble+Heuristics	0.705	0.422

Table 3: Evaluation scores of submitted system (first row) as well as other runs of our system.

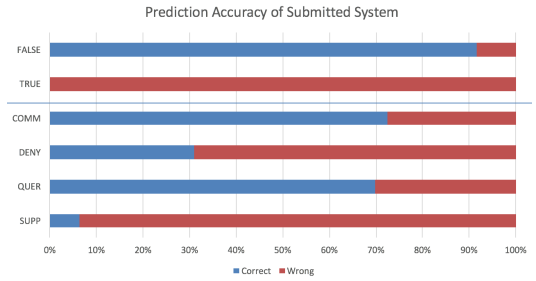

Figure 2: Prediction Accuracy of Submitted Systems in Subtask B and Subtask A

stage of assigning labels. For example, over 60% of the messages containing a question mark were queries. Hence any message containing a question mark was tagged as a query.

4 Results

Table 3 shows the results of our experiments. We submitted the MaxEnt results. However, the ensemble method (combination of all three models) shows a much better performance.

Figure 2 demonstrates the number of correct categories we classified accurately (blue bar). Our systems performed best at predicting the "comment" and "query" in subtask A and "false" in subtask B. The poor performance on "support" in subtask A and "true" in subtask B can be attributed to our post-process framework, i.e. our rules are not sufficiently discriminative. A work-around is to label all tweets as "support" and then implement the if-then rules.

5 Discussion

In this section, we briefly touch upon a few observations from our experiments. First, the actual twitter text should not be cleaned in any way, i. e., errors, misspellings, acronyms etc. contained in the text help in the task. Using rule-based heuristics derived from a statistical analysis of the characteristics of the training data, helps in a post-processing step to improve the classification performance of test data.

6 Conclusion

We implemented hybrid systems, i. e., combinations of statistical (classifier) and rule-based (heuristics) modules. It can be observed that textual features and metadata benefit both tasks. In terms of future work, we plan to implement a better cascading model, i. e., to assign probabilities to the heuristics.

Acknowledgments

We would like to thank the anonymous reviewers for their insight and helpful comments on our first draft. The project Digitale Kuratierungstechnologien (DKT) is supported by the German Federal Ministry of Education and Research (BMBF), Unternehmen Region, instrument Wachstumskern-Potenzial (No. 03WKP45). More information on the project can be found online at `http://www.digitale-kuratierung.de`.

References

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. Semeval-2017 task 8: Rumoureval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, pages 60–67. http://www.aclweb.org/anthology/S17-2006.

Eibe Frank and Remco R. Bouckaert. 2006. Naive Bayes for text classification with unbalanced classes. In *Proc 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*. Springer, Berlin, Germany, pages 503–510.

Nick Littlestone. 1988. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning* 2(4):285–318. https://doi.org/10.1023/A:1022869011914.

Robert Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING-02, pages 1–7. https://doi.org/10.3115/1118853.1118871.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Julia Minguillon. 2002. *On Cascading Small Decision Trees*. Ph.D. thesis, Universitat Autonoma de Barcelon.

Georg Rehm and Felix Sasaki. 2015. Digitale Kuratierungstechnologien – Verfahren für die effiziente Verarbeitung, Erstellung und Verteilung qualitativ hochwertiger Medieninhalte. In *Proceedings der Frühjahrstagung der Gesellschaft für Sprachtechnologie und Computerlinguistik (GSCL 2015)*. Duisburg, pages 138–139. 30. September–2. Oktober.

Georg Rehm and Felix Sasaki. 2016. Digital Curation Technologies. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation (EAMT 2016)*. Riga, Latvia.

Arkaitz Zubiaga and Heng Ji. 2014. Tweet, but verify: epistemic study of information verification on twitter. *Social Network Analysis and Mining* 4(1):163. https://doi.org/10.1007/s13278-014-0163-y.