Bootstrapping patterns for the detection of mobility related events

Britta Grusdt* **Jan Nehring*** **Philippe Thomas***
*firstname.lastname@dfki.de
German Research Centre for Artificial Intelligence
Alt-Moabit 91 c
10559 Berlin

Abstract

This work presents a method to extract traffic events from German texts. We present a rule based system, where patterns are automatically extracted and ranked using a bootstrapping approach. These patterns are subsequently evaluated and annotated by human annotators. The resulting pattern are evaluated on three different text sources (Tweets, traffic RSS feeds, and news articles) with different language styles. Through the use of three data sets we cannot only evaluate the usefulness of the approach in a single domain but also evaluate the domain portability of the proposed approach. We further perform an error analysis to identify problems of the current system.

1 Introduction

Keeping up-to-date with the current traffic situation is a difficult task, especially in modern large city environments. In such cities a large variety of transportation options exist (*e.g.,* public or individual transport) and public transportation is often managed by more than one authority. Information about events impacting the personal travel route is therefore provided by different information providers. For example, local radio stations provide local information about traffic jams, cities provide information about planned road blocks, and the public transportation operators post information on current events. Because many different sources exist, it is difficult to keep a complete overview of the current traffic situation.

Much information on traffic related events is publicly available. For example, user are enabled to report traffic jams or accidents using social media, such as Twitter. Other text-sources are online news websites and various syndication feeds. However turning these unstructured textual data into structured information is a challenging problem.

Rule based information extraction technologies using manually crafted patterns is a common technique to extract structured information from unstructured textual data. The major drawback of this approach is that the manual creation of these patterns is a very time-consuming task. In this work we apply a technique which bootstraps the pattern generation process, sorts the patterns by their frequency and therefore and enables us to focus on the most important patterns. For example, from the sentence *Berlin: Rail replacement service between Schichauweg and Priesterweg on route S2*, we would like to extract a *Rail Replacement Service event* with the arguments *location=S2* of type *location-route*, and *start-loc=Schichauweg* and *end-loc=Priesterweg* with type *location-stop* respectively.

The scientific contributions of this paper are as follows: First, a common algorithm for pattern recognition from the biology domain is applied to the mobility domain. Second, we investigate the portability of the learned patterns by creating patterns from one dataset and evaluating them on another dataset. Third, we compare the characteristics of the traffic domain in different data sources, namely RSS feeds, tweets and online news.

2 Background

In this section we list the background work starting with bootstrapping pattern learning, followed by general related work and the *Spree* framework.

2.1 Bootstrapping pattern learning

Rule based methods to generate structured information from textual data using patterns is a common technique in information extraction. Hand crafted rules often achieve high precision, but suffer from relatively low recall (Andrade and Bork, 2000; Cohen et al., 2009). The main reason being that manual rule construction is a time and labor intensive task. Several strategies have been proposed to overcome this problem:

The distant supervision assumption (Mintz et al., 2009) can be used to automatically label a large set of document using known relationships covered in a knowledge base. This strategy has been applied by Thomas et al. (2011) to learn a large set of linguistic patterns from dependency graphs, without requiring manually labeled data. Ravikumar et al. (2012) describe a similar strategy, but implement a fuzzy dependency graph matching strategy in order to increase recall.

Caporaso et al. (2007) used patterns and rules to recognize entities in biomedical text. The proposed strategy minimizes the effort of generating rules by generating a large set of rules semi-automatically. These rules are then sorted by frequency of occurrence. Then a human annotator can annotate only the most frequent rules which results in a higher recall than manually crafted rules. The same strategy was applied by Thomas and Leser (2013) for histone modification recognition. In both domains the approach achieved high precision ($> 90\,\%$) and recall ($> 80\,\%$) values. In our work presented here we also follow this approach.

2.2 Related Work

Leveraging social media for the extraction of traffic information has been previously covered (Wanichayapong et al., 2011; Kosala et al., 2012; Schulz et al., 2013; Gong et al., 2015; Gu et al., 2016).

The work of Wanichayapong et al. (2011) focuses on road traffic information extraction from Tweets in Thailand. The authors propose a strategy to extract 3-ary traffic events. The arguments include the affected road, as well as start, and endpoint of the extracted traffic event.

Schulz et al. (2013) classify English Tweets into ”car accident related” and ”not car incident related”. Relevant messages are geolocated using Stanford Entity Recognizer (Finkel et al., 2005) and the MapQuest geocoding API. A similar path has been described by Gu et al. (2016) recognizing traffic events in the Pittsburgh and Philadelphia metropolitan area. Tweets are classified into traffic incident relevant or irrelevant and then geocoded using regular expressions and a geo-parser. Both approaches focus on the classification of Tweets and do not perform n-ary event extraction.

Pereira et al. (2013) use accident reports to predict road clearance duration for expressways in Singapore. From these reports, the authors combine features derived from structured road traffic data (*e.g.,* number of cars per lane) with features from the textual description of the current situation.

D’Andrea et al. (2015) evaluate various machine learning algorithms to classify Italian Tweets into traffic and non-traffic related. Traffic related Tweets in direct vicinity are clustered using the user provided GPS location. The results of the clustering is then compared with data from news websites. This analysis unveils that traffic related information can be often extracted faster or in similar time ranges from Tweets as provided by news websites.

Gong et al. (2015) describe a method to map geolocated Tweets to the Australian road network in order to detect congested segments. The work concentrates on the structured location data, provided in a Tweet and ignores information covered by the text itself. Leveraging geocoding techniques could potentially help to improve this method, by mapping the text to geographic features, such as cities or streets.

The GermEval 2017 conference featured a shared task on aspect-based sentiment analysis in the mobility domain (Wojatzki et al., 2017). The corpus focused on customer messages about “Deutsche Bahn” and contains 22,000 German messages from various social media sources.

In contrast to the previously described approaches, we focus on n-ary mobility event extraction from texts. We consider not only road accidents or traffic-jams, but also events in public transportation networks. Our approach extracts mobility events from different text resources (Twitter, RSS, and News) and is evaluated on a publicly available corpus (Schiersch et al., 2018). To the best of our knowledge, this is the first study of n-ary mobility events extraction from German texts.

2.3 Spree

Spree (Hennig et al., 2016) is a scalable platform for real-time, fine-grained event extraction and geospatial visualization. It processes textual data from different data sources to generate structured information from unstructured data sources. The platform builds on the big data analytics framework Apache Flink (Alexandrov et al., 2014) which allows high-throughput stream computing. Spree contains a linguistic analysis pipeline which was used to generate the patterns in this project. The details of the pattern generation strategy are covered in the methods section.

3 Mobility events

Using our linguistic pipeline we annotated different types of concepts in the data. Table 1 lists these concepts:

Concept	Description
Location-City	Municipalities, *e.g.,* cities
Location-Street	Streets, *e.g.,* highways
Location-Route	Transit routes, *e.g.,* S1
Location-Stop	Transit stops, *e.g.,* Wien Hbf.
Date or Time	Point in time, *e.g.,* today
Duration	Time periods *e.g.,* 10 minutes
Distance	Distances, *e.g.,* 250 meter
Disaster	Disasters, *e.g.,* flood
Trigger	Trigger,*e.g.,* heavy traffic

Table 1: Entities and concepts annotated in the data.

Using these concepts we want to detect the following traffic events: *Accident*, *Canceled route*, *Canceled stop*, *Delay*, *Disaster*, and *Obstruction*. Each of these events uses named entities as arguments, some are required and others optional. For example, the *Accident* event requires an exact *location* and *trigger* argument. The remaining arguments *delay-time*, *direction*, *start-location*, *end-location*, *start-date*, *end-date*, *cause* are optional. For reasons of brevity we do not list detailed argument lists and explanations of each event type. Interested readers may refer to the following publication for more details on the event definitions (Schiersch et al., 2018).

4 Methods

In this section we describe the generation of the automatically labeled corpus followed by a brief discussion of the pattern generation process and the evaluation strategy.

4.1 Corpus

For linguistic pattern generation we require a large corpus containing articles for the domain of interest. To this end, we collected traffic related tweets and RSS messages in the time period between 01.01.2016 and 01.04.2016. Tweets in German language were collected by 201 different keywords (*e.g.,* traffic jam, roadway, accident, ...) and 164 user channels (*e.g.,* public transport providers, police stations, ...). RSS messages were periodically scraped (every 15 minutes) from German traffic authorities, such as the German automobile association, German railway companies, and traffic information from local radio stations. This approach yielded in 7,155,862 tweets and 5,591,654 RSS items. Due to the periodic retrieval of RSS feeds, several items are collected multiple times. Deduplication of these RSS feeds yielded 90,604 unique RSS items. Some characteristics of the two different unlabeled corpora are shown in Table 2. The table indicates that RSS messages are approximately twice as long as average tweets.

Property	Tweets	RSS
# documents	7,155,862	90,604
# tokens	122,992,580	3,709,826
avg. tokens	17.19	40.95
avg. characters	101.57	239.68
extracted patterns	392,191	87,756
distinct patterns	348,876	28,614
annotated patterns	97	138

Table 2: Characteristics about the unlabeled corpora used for pattern generation. Averages are per document.

4.2 Pattern generation

Patterns are generated in a semi-automatic fashion by the following strategy originally proposed by Caporaso et al. (2007). Both unlabeled text resources (Twitter and RSS-feeds) are processed using the Spree architecture. This includes sentence detection (for RSS feeds only), tokenization using Stanford CoreNLP (Manning et al., 2014) and Named Entity Recognition of traffic related concepts (street, city, train-stations, and train-routes) using SPROUT (Drozdzynski et al., 2004). We extract surface patterns from sentences containing at least one location-entity (*i.e., street, train-station, train-route, or city*) and an event indicating word (*e.g.,* traffic jam, road block, ...). Each pattern consists of the shortest text span between all automatically detected location words and the event trigger. Therefore, every sentence leads to at most one pattern. Named entities are then replaced with the detected entity type. For instance, cities are replaced by <*city*>, event-triggers are replaced by <*trigger*>, and so on. The pattern generation workflow is exemplified in Figure 1. An overview of the characteristics of the extracted patterns is shown in Table 2.

4.3 Pattern annotation

Using the previously described strategy lead to 87,764 RSS- and 392,191 Twitter-patterns. The frequency distribution of the 2,500 most frequently extracted patterns of both text resources is shown in Figure 2. As expected, the frequencies of the extracted patterns follow a power law distribution. Out of 28,614 individual RSS-patterns, the 200 most frequent cover 29.9 % of all pattern occurrences. For Twitter, the 200 most frequent patterns cover 3.7 % of all pattern occurrences with a total 348,876 distinct patterns. We manually annotated the most frequently found patterns for both resources individually. This strategy resulted in 97 patterns for Twitter and 138 patterns for the RSS feeds. The last pattern we manually annotated from the Twitter/RSS corpus occurs 20 and 32 times respectively. From now we use the terms *Twitter*- and *RSS*-rules to refer to the manually annotated patterns generated from the two respective corpora.

4.4 Pattern matching

Manually annotated patterns are converted into regular expressions for the pattern matching step. During the matching phase the generic entity type *location* also matches all specific location subtypes *e.g.,* city, street. Hashtags in the annotated patterns become optional when they are converted into regular expressions which further increases the number of detected mobility events. Whenever multiple patterns match the same (or overlapping) text span, we extract one mobility event from the longest string-match only.

4.5 Evaluation

We evaluated our patterns on a corpus annotated with fine grained traffic and industry related n-ary events (Schiersch et al., 2018). The corpus consists of 2,598 manually annotated documents in German language, collected from three different text sources. The sources are social media (Twitter), traffic reports from various sources (provided as RSS feeds), and newswire HTML documents. For evaluation purposes we removed all industry related events from the corpus. As our approach does not rely on manually labeled training data, the evaluation was performed on all available data. The number of remaining relations on the test set are shown in Table 3.

Corpus	Relations
Twitter	194
RSS	467
News	41
Total	702

Table 3: Number of traffic related events in the three different evaluation corpora.

For a detected event to count as true positive the predicted event type must be equal to the gold standard event type and the predicted event span must at least be subsumed by the gold standard event span. The following event types were considered in our evaluation: *Accident, TrafficJam, Delay, Obstruction, RailReplacementService, CanceledStop, CanceledRoute*.

5 Experiments

We performed the following experiments:

First, we investigated the domain adaptability of generated patterns. To this end, we applied patterns generated from a source corpus to detect events in a different target domain. For example, we used all patterns derived from Twitter (Twitter-rules) on different target corpora. Besides testing our system on test datasets collected from Twitter and RSS feeds, we additionally evaluated our system on a collection of news articles in HTML-format that were crawled from the Internet.

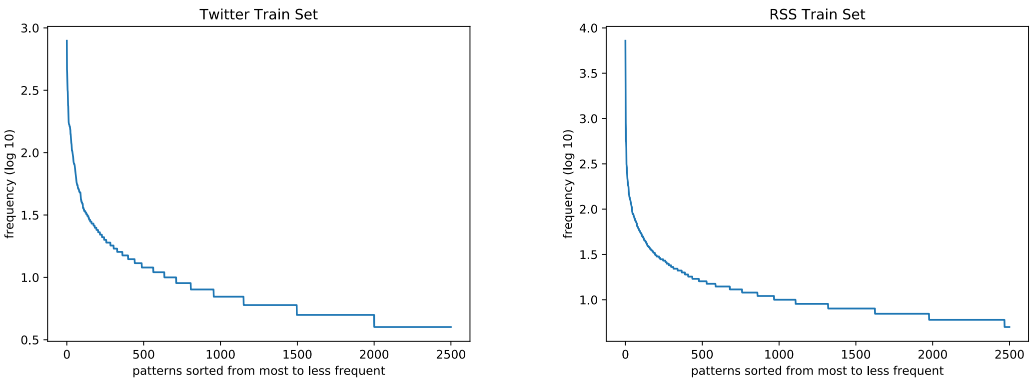
On the other hand, we were interested in how the performance of our system evolves with an increasing number of manually annotated patterns. In the best case the system should reach a steady-state, where additional patterns lead to marginal improvements only.

6 Results

The results in Table 4 indicate that the system is able to extract mobility events from texts, where patterns are extracted from the same domain. The performance on Twitter and News test data is very low for RSS-rules (F_1 of 0.099 and 0.029, respectively). This indicates that the pattern learning

	Sentence	On the A8 direction Munich - heavy accident between Dasing and Sulzmoos.
1.	NER	On the A8 direction Munich - heavy accident between Dasing and Sulzmoos.
2.	Blinding	On the <street> direction <city> - <trigger> between <city> and <city>.
3.	Pruning	<street> direction <city> - <trigger> between <city> and <city>
4.	Annotation	<location> direction <direction> - <trigger> beween <start> and <end>

Figure 1: Example of the different steps for pattern generation. First, relevant entities are recognized using SPROUT. Second, relevant terms are replaced by the respective entity type (i.e. <city>, <station>, <street>, <trigger>, ...). Third, surface pattern are generated and sorted by rank. Fourth, pattern are manually refined.



(a) Twitter-pattern distribution

(b) RSS-pattern distribution

Figure 2: Frequency of the 2,500 most frequently extracted patterns from the Twitter and the RSS corpus. Patterns are sorted by frequency rank. Please note that the y-axis uses a logarithmic scale.

must be domain dependent, *i.e.* patterns learned from one domain cannot be applied on data of another domain. For the patterns extracted from Twitter we observe a slightly different behavior. In terms of F_1 the performance is similar for Twitter and the News domain. On the out-of-domain RSS dataset we still observe a comparably low performance. However, the combination of the two sets of patterns (Twitter- and RSS-rules) can be useful to increase recall on Twitter and RSS. On the News dataset we cannot observe a positive impact on recall using the combination of patterns, as the RSS-rules extract only very few events on this type of resource. Individual results for the seven mobility-events using the combination of all patterns is shown in Table 5. The result show that some event-types (*i.e.,* Traffic jam and rail replacement service) can be extracted with high precision and recall, while some events are rather difficult to extract (*i.e.,* route and stop cancellation).

We also compared our result with the relation extraction system DARE (Xu et al., 2007; Krause et al., 2012). DARE learns minimal dependency subgraphs that connect all relation arguments. We omit any filtering of the extracted dependency patterns, i.e. we include all learned patterns, even ambiguous or low-frequency ones, in the model. Filtering the DARE patterns seems necessary for a valid comparison with our method but since the recall values for DARE are so low, it would not substantially improve the results for F1. Our approach outperforms DARE on all three corpora. We assume the main reason is the difficulty to generate dependency parse trees on the telegram style texts, which are a prerequisite for DARE.

In our second experiment, we subsequently increased the proportion of annotated patterns. The results are shown in Figure 3 and 4. The results indicate that the addition of rules has, apart from a few exceptions covered in the Discussion, a positive impact on recall and F_1. By adding more rules, some false negatives are converted to true positives (or false positives) which decreases the number of false negatives and this, in turn, results in higher recall values. Precision does not show a comparable consistent behavior. When only a few annotated rules are used, the number of false positives tends to be smaller which might result in a very high precision.

7 Discussion

In the following we will discuss aspects of our results that were different from our expectations and needed some more investigation: Regarding recall, we observe in few cases that less patterns sometimes achieve a higher recall than using more patterns. This can be observed for the RSS-rules tested on the Twitter corpus. Here, the recall is higher when using 80 % of the annotated patterns than using all available patterns. Since the recall is the ratio of true positives with respect to the sum of true events, it should not drop when more rules are included. It should rather increase the number of TPs or decrease the number of FNs yielding equal or greater recall values. In our system, it is possible that some input text will be matched by multiple rules. In that case, we decided to use the rule that matches the longest span in the input text. In rare cases this strategy leads to the introduction of false negatives. For example, the two rules shown in Figure 5 both match the following example sentence:

RB 40 Niedersachsen Zugausfall und Ersatzverkehr Braunschweig Hbf Magdeburg:

- (1) **Ersatzverkehr Braunschweig Hbf Magdeburg**
- (2) **Zugausfall und Ersatzverkehr Braunschweig Hbf Magdeburg**

For rule (1), the trigger-word that determines the event type, is **Ersatzverkehr** (RailReplacementService) whereas for rule (2), it is **Zugausfall** (train cancellation). When the true event is RailReplacementService which is only induced by (1), the choice of rule (2) results in a false negative (RailReplacementService) as well as in one false positive, namely the Delay event induced by rule (2). This explains why the recall sometimes drops when more annotated rules are included into our system.

We previously showed in Table 4 that the combination of Twitter- and RSS-rules is beneficial on recall. We evaluated the overlap between RSS- and Twitter-rules in Figure 6. The intersection between the two different set of rules is with two rules relatively small. This indicates that the rules extracted from the two different datasets cover different linguistic phenomena. Therefore, the combination of both rule-sets increases recall on several evaluation corpora.

	Twitter			RSS			News		
	P	R	F_1	P	R	F_1	P	R	F_1
Twitter	0.7209	0.2719	0.3949	0.4588	0.0835	0.1413	**0.6800**	**0.2615**	**0.3778**
RSS	**0.8571**	0.0526	0.0992	**0.7801**	0.3191	0.4529	0.3333	0.0154	0.0294
Merged	0.7263	**0.3026**	**0.4272**	0.7458	**0.3769**	**0.5007**	**0.6800**	**0.2615**	**0.3778**
DARE	0.3750	0.1098	0.1698	0.2222	0.0699	0.1063	0.0000	0.0000	0.0000

Table 4: Evaluation results for the different set of patterns on the different evaluation corpora. The left column indicates the source of the patterns. Patterns have been evaluated on Twitter, RSS, and News documents. The highest result is highlighted in boldface.

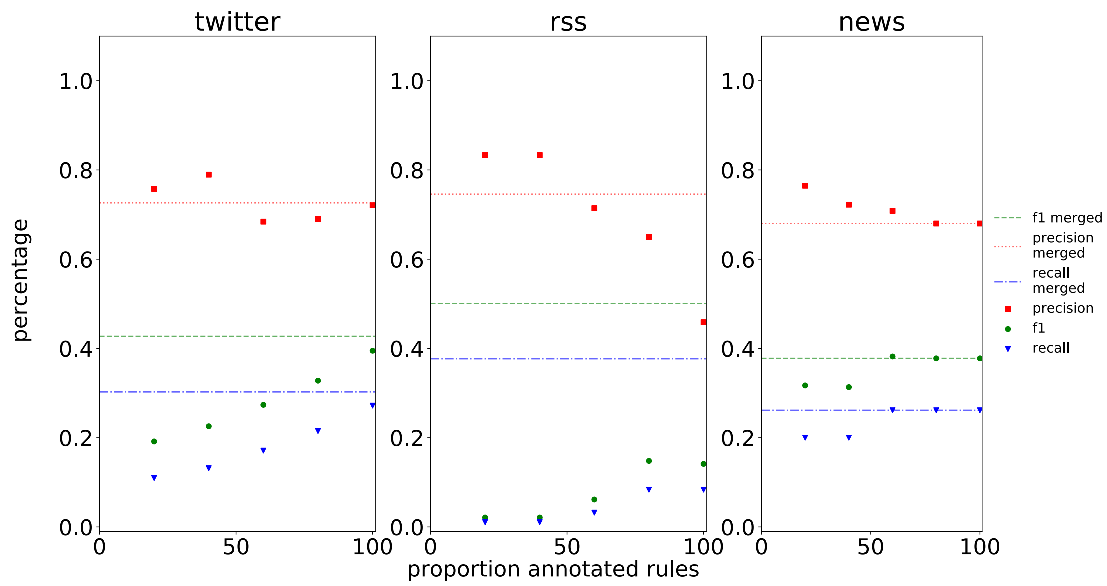


Figure 3: Precision, Recall and F_1-score when the system uses Twitter rules and test data from twitter (left), RSS (middle) or news (right) corpora. The horizontal lines represent the results when the merged RSS and Twitter rules are used.

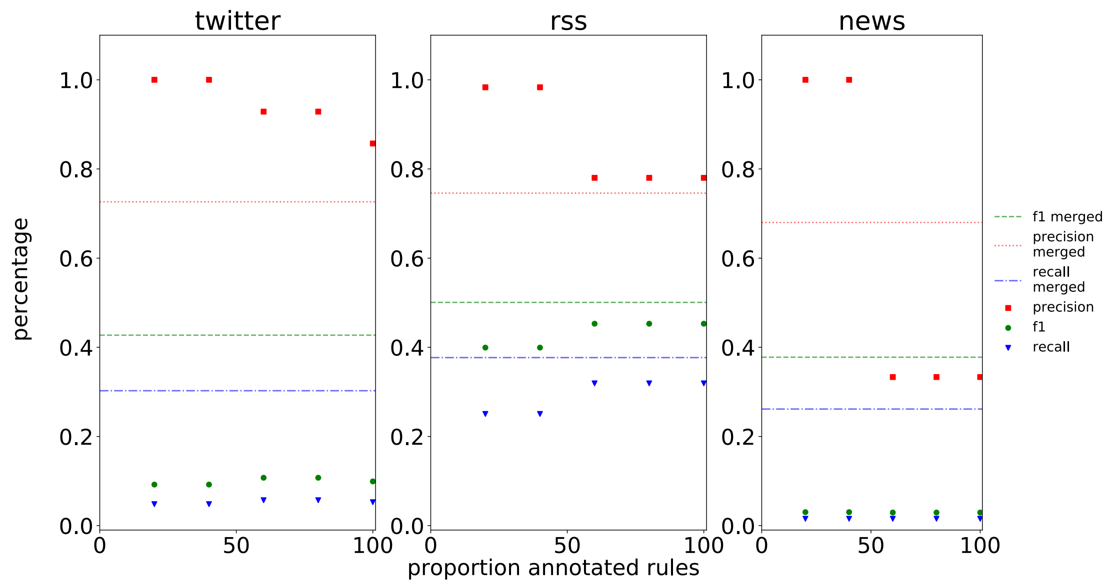


Figure 4: Precision, Recall and F_1-score when the system uses RSS rules and test data from twitter (left), RSS (middle) or news (right) corpora. The horizontal lines represent the values that we get when the system uses the combination of RSS and Twitter rules.

Relation-Type	P	R	F_1
Accident	0.83	0.54	0.66
Canceled route	0.00	0.00	0.00
Canceled stop	0.00	0.00	0.00
Delay	0.73	0.46	0.57
Obstruction	0.58	0.26	0.35
Rail replacement service	1.00	0.88	0.93
Traffic jam	0.99	0.73	0.84

Table 5: Performance of event detection using the combination of all patterns on all seven sub-event types.

```
(1) TRIGGER[relationType] LOCATION–STOP
LOCATION–STOP

(2) TRIGGER[relationType] und TRIGGER
LOCATION–STOP LOCATION–STOP
```

Figure 5: Two annotated RSS-rules. *relationType* designates the trigger inducing the event type.

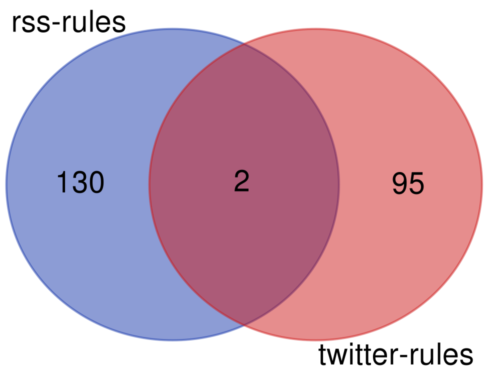


Figure 6: Venn diagram for RSS- and Twitter-rules.

The bottleneck that keeps the F_1-scores at a relatively low value is recall since there is a quite high number of false negatives. We use our system for a web application visualizing the extracted traffic events. By combining different sources, from which the events are extracted in real time, we hope to provide the user with reliable traffic information faster than it is possible with information available from radio stations or other sources. We assume that every traffic event appears multiple times, e.g. as tweet from multiple users who describe the event in different words. Since our evaluation is not on the semantic level of events, but sentence based, the recall values we obtain remain low. Therefore we focus on correctly predicting traffic-related events (i.e. precision) in the first place instead of recall by taking a closer look at the events that were classified as false positives. Our investigations revealed that a small number of the annotations were incorrect. Furthermore, we observed that some sentences are annotated with overlapping annotations. For instance, the example described above was annotated twice, once as RailReplacementService and once as Delay. Due to the fact that our system detects at most one event for the same text span, this necessarily results in at least one false negative.

In the following, we will discuss the problems we observed, when we automatically extract the patterns which are then used for manual annotations.

All extracted patterns are based on entire sentences and since Tweets and RSS feeds are often formulated in a telegram style, it can be difficult to detect sentence boundaries correctly for them. This may result in extracted patterns whose spans exceed the correct sentence boundaries and the longer a detected pattern is, the less likely it becomes to find matches for it. However, we assume that important subpatterns of these long patterns, *i.e.* those appearing frequently, are extracted elsewhere.

Also sometimes the information spans multiple sentences. E.g. *Alert for highway A7: There is a traffic jam.* Since our system extracts events from single sentences only it cannot detect events of this type.

Other problems arise from the automatic annotation of concepts by the linguistic pipeline. Often words have overlapping annotations, e.g. the highway interchange *Kreuz Köln* contains the name of a city in its name. In this case the word *Köln* is annotated twice, once as an interchange and once as a city. The pattern algorithm does not work with overlapping annotations. In this case we chose to use only the concept with the longest span and to remove other concepts.

8 Conclusion

We successfully applied a bootstrapping strategy to automatically build a large set of patterns describing potential mobility related events. Patterns have been derived from two different types of text resources. In our analysis, we observe that patterns derived from Twitter are well suited for short messages from social media, as well as longer news articles. RSS feeds describing traffic event seem to be different from the other two resources, as patterns derived from RSS have extremely low recall values on Twitter and News feeds. In following work we would like to tackle the question how the annotated patterns can be transferred to other target languages.

Acknowledgements

This research was partially supported by the German Federal Ministry of Economics and Energy (BMWi) through the project SD4M (01MD15007B) and by the German Federal Ministry of Transport and Digital Infrastructure through the project DAYSTREAM (19F2031A). Further, we want to thank Leonhard Hennig for very helpful discussions and his advice concerning DARE.

References

Alexander Alexandrov, Rico Bergmann, Stephan Ewen, Johann-Christoph Freytag, Fabian Hueske, Arvid Heise, Odej Kao, Marcus Leich, Ulf Leser, Volker Markl, Felix Naumann, Mathias Peters, Astrid Rheinländer, Matthias J. Sax, Sebastian Schelter, Mareike Höger, Kostas Tzoumas, and Daniel Warneke. 2014. The stratosphere platform for big data analytics. *VLDB J.*, 23(6):939–964.

Miguel A. Andrade and Peer Bork. 2000. Automated extraction of information in molecular biology. *FEBS Letters*, 476(1):12 – 17. Birmingham Issue.

J. Gregory Caporaso, William A. Baumgartner, David A. Randolph, K. Bretonnel Cohen, and Lawrence Hunter. 2007. Rapid pattern development for concept recognition systems: Application to point mutations. *Journal of Bioinformatics and Computational Biology*, 05(06):1233–1259.

K. Bretonnel Cohen, Karin Verspoor, Helen L. Johnson, Chris Roeder, Philip V. Ogren, William A. Baumgartner, Jr., Elizabeth White, Hannah Tipney, and Lawrence Hunter. 2009. High-precision biological event extraction with a concept recognizer. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, BioNLP '09, pages 50–58, Stroudsburg, PA, USA. Association for Computational Linguistics.

Eleonora D'Andrea, Pietro Ducange, Beatrice Lazzerini, and Francesco Marcelloni. 2015. Real-time detection of traffic from twitter stream analysis. *IEEE transactions on intelligent transportation systems*, 16(4):2269–2283.

Witold Drozdzynski, Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer, and Feiyu Xu. 2004. Shallow processing with unification and typed feature structures — foundations and applications. *Künstliche Intelligenz*, 1:17–23.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 363–370, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yikai Gong, Fengmin Deng, and Richard O. Sinnott. 2015. Identification of (near) real-time traffic congestion in the cities of australia through twitter. In *Proceedings of the ACM First International Workshop on Understanding the City with Urban Informatics*, UCUI '15, pages 7–12, New York, NY, USA. ACM.

Yiming Gu, Zhen (Sean) Qian, and Feng Chen. 2016. From twitter to detector: Real-time traffic incident detection using social media data. *Transportation Research Part C: Emerging Technologies*, 67:321 – 342.

Leonhard Hennig, Philippe Thomas, Renlong Ai, Johannes Kirschnick, He Wang, Jakob Pannier, Nora Zimmermann, Sven Schmeier, Feiyu Xu, Jan Ostwald, and Hans Uszkoreit. 2016. Real-time discovery and geospatial visualization of mobility and industry events from large-scale, heterogeneous data streams. In *Proceedings of ACL-2016 System Demonstrations*, pages 37–42. Association for Computational Linguistics.

Raymondus Kosala, Erwin Adi, and Steven . 2012. Harvesting real time traffic information from twitter. 50:1–11.

Sebastian Krause, Hong Li, Hans Uszkoreit, and Feiyu Xu. 2012. Large-Scale Learning of Relation-Extraction Rules with Distant Supervision from the Web. In *Proc. of ISWC*, pages 263–278.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1003–1011, Stroudsburg, PA, USA. Association for Computational Linguistics.

Francisco C. Pereira, Filipe Rodrigues, and Moshe Ben-Akiva. 2013. Text analysis in incident duration prediction. *Transportation Research Part C: Emerging Technologies*, 37:177 – 192.

KE Ravikumar, Haibin Liu, Judith D. Cohn, Michael E. Wall, and Karin Verspoor. 2012. Literature mining of protein-residue associations with graph rules learned through distant supervision. *Journal of Biomedical Semantics*, 3(3):S2.

Martin Schiersch, Veselina Mironova, Maximilian Schmitt, Philippe Thomas, Aleksandra Gabryszak, and Leonhard Hennig. 2018. A german corpus for fine-grained named entity recognition and relation extraction of traffic and industry events. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Axel Schulz, Petar Ristoski, and Heiko Paulheim. 2013. I see a car crash: Real-time detection of small scale incidents in microblogs. In *The Semantic Web: ESWC 2013 Satellite Events*, pages 22–33, Berlin, Heidelberg. Springer Berlin Heidelberg.

Philippe Thomas and Ulf Leser. 2013. HistoNer: Histone modification extraction from text. In *Proceedings of BioLINK Special Interest Group*, pages 52–55.

Philippe Thomas, Stefan Pietschmann, Illés Solt, Domonkos Tikk, and Ulf Leser. 2011. Not all links are equal: Exploiting Dependency Types for the Extraction of Protein-Protein Interactions from Text. In *Proceedings of BioNLP 2011 Workshop*, pages 1–9, Portland, Oregon, USA.

Napong Wanichayapong, Wasawat Pruthipunyaskul, Wasan Pattara-atikom, and Pimwadee Chaovalit. 2011. Social-based traffic information extraction and classification. In *11th International Conference on ITS Telecommunications (ITST)*, pages 107–112.

Michael Wojatzki, Eugen Ruppert, Sarah Holschneider, Torsten Zesch, and Chris Biemann. 2017. GermEval 2017: Shared Task on Aspect-based Sentiment in Social Media Customer Feedback. In *Proceedings of the GermEval 2017 – Shared Task on Aspect-based Sentiment in Social Media Customer Feedback*, pages 1–12, Berlin, Germany.

Feiyu Xu, Hans Uszkoreit, and Hong Li. 2007. A Seed-driven Bottom-up Machine Learning Framework for Extracting Relations of Various Complexity. In *Proc. of ACL*, pages 584–591.