What Is Good for One City May Not Be Good for Another One: Evaluating Generalization for Tweet Classification Based on Semantic Abstraction

Axel Schulz[1] and Frederik Janssen[2]

[1] Technische Universität Darmstadt
Telecooperation Lab
aschulz@tk.informatik.tu-darmstadt.de
[2] Knowledge Engineering Group
janssen@ke.tu-darmstadt.de

Abstract. Social media is a rich source of up-to-date information about events such as incidents. The sheer amount of available information makes machine learning approaches a necessity. However, those most often are focused on regionally restricted datasets such as data from only one city. The important fact that social media data such as tweets varies considerably across different cities is neglected. To cope with this problem, usually the data of each city needs to be labeled, which is costly and time consuming.

To omit such an expensive labeling procedure, another idea is to train a general model on one city and then apply it on data of a different city. In this paper, we present *semantic abstraction* that relies on features derived from Linked Open Data as well as location and temporal mentions. We show that it is a valuable means for such a generalization (increase of F-measures by 8.24% and 7.32%, respectively). Furthermore, to get a thorough understanding of the generalization problem itself, we conducted an in-depth evaluation of our approach based on considering rule-based models. By examining the learned rule sets, we can conclude that a feature selection by an expert seems to be necessary especially for the Linked Open Data features.

1 Introduction

Social media platforms are widely used for sharing information about incidents. Due to the large amount of data that is created every day, automatic filtering is a necessity. Many approaches use machine learning to classify the type of an incident mentioned in social media such as tweets [1], [16]. This way, social media data became a valuable source of timely incident information. However, to build these classifiers labeled data is required. Given the large quantity of data in this context, creating such annotations is time consuming and hence costly. Also, and more importantly, datasets most often are naturally restricted to a certain context, i.e., labeling the data is only valid for one city.

Obviously, the motivation behind this is to find the best classifier for exactly this problem. However, such a classifier does not generalize to other regions, i.e., other cities, because the text in tweets has special properties compared to structured textual information. The expectation is that a model learned on one city consequently works well on that city as similar words are used, but not necessarily on data from a different city. Named entities used in texts are likely to be related to the location where the text was created or contain certain topics. Thus, when the classifier relies on named entities that are unique in the given city such as street names etc., it is not suited for other cities where these do not occur. These aspects complicate the task of generalizing a classification model in the domain of social media texts. As an example consider the following two tweets shown in Listings 1.1 and 1.2.

Listing 1.1. Example tweet containing temporal expressions and a location mention.

```
RT: @People 0noe friday afternoon in heavy traffic, car
    crash on I-90, right lane closed
```

Listing 1.2. Example tweet containing a location mention.

```
Road blocked due to traffic collision on I-495
```

Though both tweets describe an incident, the similarity between the texts is not easily extracted using standard bag of words features. Nevertheless, both tweets consist of entities that might describe the same thing with different wording. In this example, ”accident” and ”car collision” are similar expressions for the same type of event. Furthermore, ”I90” and ”I-495” are both names of streets. With simple syntactical text similarity approaches it is not easily possible to make use of this semantic similarity, though, it definitely is valuable for classifying both tweets.

We tackle this problem by creating a generalized model using training information in form of social media data collected in one city to classify data that stems from a different city. In contrast to traditional Feature Augmentation [6] prior to model creation features are not discarded but abstracted to city-independent ones in our approach. To do so, automatic named entity and temporal expression recognition is used to introduce abstract features based on occurrence of location and temporal mentions. Additionally, background information provided by Linked Open Data[1] (LOD) is used to obtain new features that are universally applicable. This is done by scanning our dataset for named entities and enhancing the feature space with the direct types and categories of the entities at hand.

In an evaluation on two datasets we show that the novel approach for semantic abstraction improves classification on each of them. It also is valuable to generalize the model when it is trained on one city but applied on a different one. Furthermore, we conducted an in-depth analysis of the rule-based models we employed, which enables us to easily identify what rule was used to classify the example at hand, what conditions this rule consists of and, consequently, what features were used. This analysis shows that (1) features generated by semantic abstraction are frequently used and that (2) LOD features that are assumed to be the most general ones interestingly even tend to decrease classification performance when used in a generalized model.

After the introduction, we present the approach for semantic abstraction of social media data in Section 2, followed by a description of our dataset in Section 3. Next, we

[1] `http://linkeddata.org/`

present the results of our evaluation (see Section 4) followed by an overview of related work in Section 5. Finally, we close with a conclusion and future work (cf. Section 6).

2 Named Entity and Temporal Expression Recognition on Unstructured Texts

We apply several steps to enable semantic abstraction. We first identify entities and expressions to use their attributes as features. To do this, first, Named Entity Recognition (NER) has to be performed to extract these named entities. As there is no common agreement on the definition of a named entity in the research community, we use the following definitions throughout this paper:

Definition 1. *An* ***entity*** *is a physical or non-physical thing that can be identified by its properties (e.g., United Kingdom, Seattle, my university). A* ***named entity*** *is an entity that has been assigned a name ("Technische Universität Darmstadt"). Thus, the mention of a named entity in a text is defined as* ***named entity mention****.*

We further distinguish named entities of type location:

Definition 2. *A* ***proper location mention*** *(also called* ***toponym****) is defined as the named entity mention of a location. Typically, a location mention is a proper name (represented by a noun or noun phrase) that is given to a location. In contrast, we define* ***common location mentions*** *as location mentions for which no indication of the name is given in a text.*

In natural language names are not necessarily unique and therefore have to be disambiguated. E.g., there are 23 cities in the USA that are named "Paris". This means that named entities may only be unique within the appropriate context, but due to the nature of short texts, often this contextual information is missing [13]. However, as our prior work in tweet geolocalization showed [15] the combination of different information sources helps coping with the disambiguation problem. In this paper, we think that the combination of different features is valuable too.

Temporal expressions are another important part of short texts and therefore should be used as features. In this paper, we do not treat them as named entities in the sense as we defined these. Thus, beside NER we apply Temporal Expression Recognition and Normalization (TERN). TERN copes with detecting and interpreting temporal expressions to allow further processing. Following the definition of [2], we define temporal expressions as follows:

Definition 3. *We define* ***temporal expressions*** *as tokens or phrases in text that serve to identify time intervals. E.g., "yesterday", "last Monday", "05.03.2013", "2 hours".*

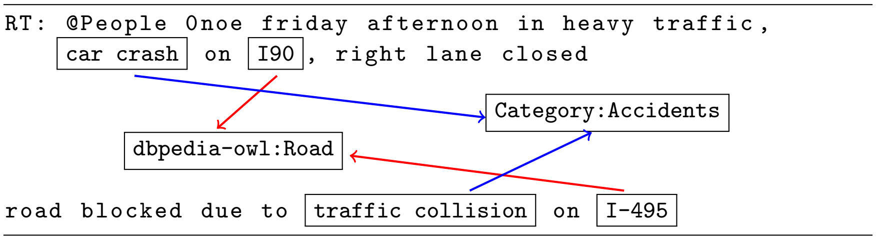
Now, we apply different approaches for identifying and classifying named entities and temporal expressions in tweets. For both types we adapted several frameworks to use the results as features for our semantic abstraction approach. First, we use Linked Open Data (LOD) as a source of interlinked information about various types of entities. Second, NER is applied for extracting location mentions and third, we adapted a framework for temporal expression extraction.

2.1 Named Entity Recognition and Replacement using Linked Open Data

As a first approach, we use Linked Open Data (LOD) as a source of interlinked information about entities to generate new features. For instance, different named entity mentions in social media texts are used synonymously to refer to the same entity. ”NYC”, ”New York City”, or ”The Big Apple” all refer to the same city *New York City*. With simple text similarity measures, this relationship is not directly visible. However, as all mentions relate to the same URI in DBpedia, this background knowledge about an entity may be used as feature. Another example are the proper location mentions ”Interstate-90” and ”Interstate-495” for which both URIs in DBpedia are linked to the same type ”dbpedia-owl:Road”. Also this example shows that semantic similarity between named entity mentions or rather the relationship between entities can be identified using LOD.

In Listing 1.3 two shared relations are shown for both example tweets. However, the extraction of this information is not easily achieved. First, named entity mentions have to be extracted. Second, they have to be mapped to the corresponding URIs, which makes disambiguation of them a necessity. Third, the valuable relations have to be identified and obtained. In order to generate features from tweets, we use *DBpedia Spotlight* [9] for the first two steps. In Section 3.2, we show how features are generated based on these URIs.

Listing 1.3. Extracted DBPedia properties for two tweets showing semantic similarity.


RT: @People 0noe friday afternoon in heavy traffic,
car crash on I90, right lane closed
Category:Accidents
dbpedia-owl:Road
road blocked due to traffic collision on I-495

2.2 Location Mention Extraction and Replacement

We make use of location mentions as another type of named entity that could be valuable as additional features for text classification. As location mentions are not easily extracted with Spotlight or often for these entities URIs are missing in DBpedia, we use a different approach for detecting them. We focus on extracting proper location mentions as well as common location mentions. Especially the later ones are used rather frequently in incident-related tweets. For instance, during our analyses we often encounter geospatial entities such as ”lane”, ”highway”, or ”school”.

For instance, in the example tweet in Listing 1.3, ”I-90” is contained, which is a proper location mention. Also ”right lane” is contained, which is a common location mention. With our approach, we recognize these location mentions. This includes different named entities such as streets, highways, landmarks, or blocks. These mentions are detected and replaced with a general annotation ”ProperLOC”. We also detect common location mentions and replace them with a general annotation ”CommonLOC”.

For location mention extraction and replacement, we use the Stanford Named Entity Recognizer[2]. The model was retrained based on 800 manually labeled tweets containing location mentions drawn from our two datasets (see Section 3.1), providing more than 90% precision. The resulting model was applied to detect location mentions for both datasets for feature generation (see Section 3.2). Compared to the LOD approach, which makes use of a generic source of background information, our approach for location mention extraction is explicitly trained for our datasets, thus, much less generalizable but much more precise.

2.3 Temporal Expression Recognition and Normalization on Unstructured Text

Finally, we extracted temporal expressions from tweets. For example, the tweet shown in Listing 1.3 contains the temporal expression ”friday afternoon” that refers to the day when an accident occurred.

For identifying temporal expressions in tweets, we adapted the HeidelTime [18] framework. The HeidelTime framework mainly relies on regular expressions to detect temporal expressions in texts. As the system was developed for large text documents with formal English language, it is unable to detect some of the used temporal expressions in the unstructured texts. Hence, as a first step, we use a dictionary for resolving commonly used abbreviations and slang (see Section 3). As a second step, we use an extension of the standard HeidelTime tagging functionality to detect temporal expressions such as dates and times. The detected expressions are then replaced with two annotations: ”DATE” and ”TIME”.

3 Generation and Statistics of the Data

In the following, we describe how the data was collected and preprocessed. Then, to get a better understanding of the data, some statistics are presented.

3.1 Data Collection

We decided to focus on tweets as a suitable example for unstructured textual information shared in social media. Furthermore, we do classification of incident-related tweets, as this type of event is common for every city and not bound to a certain place. In the following, we focus on a two-class classification problem, differentiating new tweets as ”incident related” or ”not incident related”.

As ground truth data, in November 2012 we collected 6M public tweets in English language using the Twitter Search API. For the collection, we used a 15km radius around the city centers of Seattle, WA and Memphis, TN. We focus on these two cities, as they have a huge regional distance. Also, the number of incident-related tweets is sufficiently high. We first identified and selected tweets mentioning incident-related keywords as shown in [16]. Based on these incident-related keywords, we filtered the datasets.

[2] http://nlp.stanford.edu/software/CRF-NER.shtml

As manual labeling is expensive and we needed high-quality labels for our evaluation, we selected a small subset of tweets. Furthermore, we wanted to have a different class distribution for every city. Hence, we randomly selected 500 Memphis-tweets and 1,000 Seattle-tweets containing at least one incident keyword and 1,000 Memphis-tweets and 1,500 Seattle-tweets with no incident keyword. Based on the resulting set, we removed all re-tweets, other redundant tweets, and tweets with no textual content. These tweets were manually examined by five researchers using an online survey. To assign the final coding, all coders had to agree 75% on a label. In the case of disagreement, issues were resolved in a group discussion. This gave us two datasets for our evaluation:

- **MEMPHIS** 1,082 tweets (361 incident related, 721 not incident related)
- **SEATTLE** 2,204 tweets (800 incident related, 1404 not incident related)

Though the number of tweets seems to be rather low, even without semantic abstraction the overall number of features is rather high with more than 39K for SEATTLE and 20K for MEMPHIS.

3.2 Preprocessing and Feature Generation

Before making use of our datasets, we needed to convert the texts into a structured representation so it could be used for feature generation. As a first step, the text was converted to Unicode as some tweets contain non-Unicode characters. Second, as shown before, users tend to use abbreviations. To detect commonly used abbreviations, we created a dictionary based on the data provided by the Internet Slang Dictionary & Translator[3]. Then, we identified abbreviations in tweets and replaced them with the corresponding word. Third, URLs were replaced with a common token "URL". As a next step, stopwords were removed. We also replaced digits with a common token "D". Based on the resulting text, we conducted tokenization. Thus, the text was divided into discrete words (*tokens*) based on different delimiters such as white spaces. Every token was then analyzed and non-alphanumeric characters were removed or replaced. Finally, lemmatization was applied to normalize all tokens.

After finishing the initial preprocessing steps, we extracted several features from the tweets that were used for training. The general pipeline consists of the following steps[4]: First, we make use of word-3-grams, thus, a tweet is represented as a set of words. As features we use a vector with the frequency of each n-gram. Second, we calculate the TF-IDF scores for each token [8]. We also add the accumulated TF-IDF score for each tweet as an additional feature. Third, we add syntactic features such as the number of explanation marks, questions marks, and the number of upper case characters.

The resulting feature set was further enhanced with our three semantic abstraction approaches. The different approaches were performed on the original tweet, not the preprocessed one. To enrich our feature space with semantic abstraction, we first used the RapidMiner Linked Open Data extension [12] (the *LOD* feature group). The extension

[3] `http://www.noslang.com/`

[4] For all features, additional experiments have shown that these combinations worked best.

proceeds by recognizing entities based on DBPedia Spotlight [10] to get likely URIs of the detected named entities. Then, these URIs are used to extract the types and categories of an entity. E.g., for the location mention ”I-90”, a type would be *dbpedia-owl:ArchitecturalStructure* and a category *category:Interstate_Highway_System*. In contrast to previous works, we do not treat the extracted features as binary, but use them as numeric features for our evaluation. Thus, for each tweets, the feature encodes the number of words with the same URI. Furthermore, as we only have a small number of features compared to the huge number of text features in the original dataset a feature selection was not conducted at this point of time.

Second, we used our location mention extraction approach and replaced location mentions in the unprocessed tweet texts. Based on this, the preprocessing was applied. Thus, location mentions were represented as TF-IDF features as well as word-n-grams. Furthermore, we counted the number of location mentions in a tweet. In combination, this results in a group of features for location mentions (*LOC* feature group). The same mechanism was applied to the temporal mentions, resulting in additional TF-IDF features, word-n-grams, as well as the number of temporal mentions in a tweet (*TEMP* feature group). For our evaluation we also provide the *ALL* feature group, which is the combination of the LOD, LOC, and TEMP feature groups.

3.3 Statistics

As we aimed to use two heterogeneous datasets from two cities, we analyzed how similar they are. Table 1 shows the overall number of unique tokens before and after preprocessing. The results indicate that after preprocessing, 28% of all commonly shared tokens are present in the Seattle dataset and 48% in the Memphis dataset. This shows that there are indeed huge differences between the tokens of both cities. This emphasizes the initial hypothesis that using plain n-grams is not sufficient for achieving high classification results on such diverse datasets. Furthermore, the results show the importance of applying preprocessing to get a common base of tokens for feature generation.

Table 1. Number of tokens both datasets have in common.

	Unprocessed	Processed
Seattle	10339	3606
Memphis	5657	2070
Seattle $\cap$ Memphis	1993	1007

In Table 2 the number of tweets for which location and temporal mentions, as well as LOD features could be extracted is shown. The results indicate that location mentions and LOD features could be extracted for about 50% of all tweets in both datasets. Furthermore, temporal mentions could be identified in only 20%. The table also shows that for more than 37% of all tweets location mentions as well as LOD features could be extracted in one tweet. This is likely to be a result that location mentions are also linked to URIs using Spotlight. Taking also temporal mentions into account reduces the number significantly.

Table 2. Number of tweets containing location and temporal mentions as well as LOD types and categories.

	Seattle	Memphis		Seattle	Memphis
LOC	1295 (58.76)%	522 (48.24%)	ALL	160 (7.26%)	106 (9.80%)
TEMP	403 (18.28%	265 (24.49%)	LOC + TIME	256 (11.62%)	140 (12.94%)
Types	1269 (57.58%)	566 (52.31%)	LOC + LOD	873 (39.61%)	409 (37.80%)
Categories	1222 (55.44%)	548 (50.65%)	TIME + LOD	254 (11.52%)	161 (14.88%)

Furthermore, we analyzed the number of distinct types and categories that could be extracted for both datasets (see Table 3). Comparing the LOD features for both cities shows that 880 types and 1553 categories are shared by both datasets. This means that LOD features are indeed helpful, but still a feature selection seems to be necessary.

Table 3. Number of distinct types and categories extracted for both datasets.

	Seattle	Memphis
Distinct Types	3037	1553
Distinct Categories	4812	2042
Seattle ∩ Memphis Types	880	880
Seattle ∩ Memphis Categories	1553	1553

We also analyzed the five most representative LOD features for both classes in both datasets. The representativeness was calculated based on the number of incident-related and not incident-related tweets containing a certain LOD feature. On the one hand, the results in Table 4 indicate that mostly types and categories related to location mentions are relevant for incident-related tweets. As shown, both datasets have many of these LOD features in common. On the other hand, a variety of different LOD features are present for tweets not related to incidents. In this case, both datasets have a very limited number of LOD features in common.

4 Evaluation

In the following, we present our evaluation results. We first introduce the metrics and methodology used for our evaluation. Second, we show our results when the classifier is evaluated on one city, and third we present results when training and testing is performed on data from different cities. Fourth, we analyze first approaches for optimizing the usage of LOD features for the two-cities classification problem.

Metrics and Methodology: To evaluate the learned rule sets we used one run of a ten-fold cross validation, whenever no test set was present (i.e., the cases when we evaluate on tweets of a single city). All estimates provided are F-measure values (F) as this metric is commonly used for evaluating text classifiers. The different features are combined and evaluated using the machine learning library Weka and the Ripper rule learner (JRip) algorithm [19]. We decided to use a rule learning algorithm to be able to

Table 4. The most representative LOD features for incident-related and not incident-related tweets in each dataset.

Seattle	Memphis
Incident related	
../ontology/ArchitecturalStructure	../ontology/Place
../ontology/Infrastructure	../ontology/Infrastructure
../ontology/RouteOfTransportation	../ontology/ArchitecturalStructure
../ontology/Road	../ontology/Road
../resource/Category:Interstate_5	../ontology/RouteOfTransportation
...	...
../class/yago/YagoLegalActorGeo	../class/yago/Conveyance103100490
../class/yago/YagoPermanentlyLocatedEntity	../ontology/MeanOfTransportation
../class/yago/YagoLegalActor	../ontology/Automobile
../ontology/Agent	../resource/Category:Living_people
../class/yago/Abstraction100002137	../class/yago/Instrumentality103575240
Not incident related	

interpret the resulting models. As our primary interest is to evaluate the importance of the semantic abstraction, we are not interested in finding the model that yields the highest F-measure. Albeit statistical models might have a better performance than symbolic ones, they are not interpretable and therefore not applicable for our purpose.

The first step towards the best generalization certainly is to get a thorough understanding of the abstracted features. Only then, one should proceed with tuning the models. Hence, our first experiments were conducted with an unpruned version of JRip as in this setting we most certainly will end up with many rules that consequently will have many conditions. Also, preliminary experiments show that the difference in F-measure for the pruned and unpruned versions of JRip are not substantial.

4.1 Using tweets from one city only

In the first experiment, we wanted to see how important semantic abstraction is when we use data from one city only. As we implicitly follow two goals, namely to generalize to unseen data from the same city and to generalize to a completely different city, we start by giving results for one city. In Table 5, the results for applying different feature combinations on both datasets are shown. The results for MEMPHIS show that using all features results in the best classification performance (F = 85.80%). Compared to not using semantic abstraction (NO Concepts, F = 83.66%), we get an increase of 2.14%. However, the results on this dataset also show that using only temporal features or LOD categories decreases the classification results.

For the SEATTLE dataset we get an increase of 1.94% using semantic abstraction. In this case, except the combination of LOD and temporal features, all feature combinations improve the classification results. It seems that semantic abstraction is indeed a valuable means for classification of datasets derived from one city and that a combination of all features works best.

Table 5. F-Measures for training and testing on one dataset using 10f-CV.

	MEMPHIS	SEATTLE		MEMPHIS	SEATTLE
ALL	**85.80%**	**81.17%**	LOD CATEGORIES	83.48%	79.19%
LOC TIME	85.65%	80.52%	TIME	82.45%	79.40%
LOD TIME	85.23%	78.75%	LOC	84.60%	79.47%
LOD LOC	85.26%	81.32%	NO Concepts	83.66%	79.23%
LOD	85.42%	79.40%	Majority class	53.30%	49.58%
LOD TYPES	85.33%	79.40%			

4.2 Generalizing from one city to another one

The classification results for training a classifier on one city and applying it on the other city are shown in Table 6. They indicate that using semantic abstraction outperforms the simple approach without semantic abstraction by 8.24% and 7.29%, respectively. However, training a model on SEATTLE and applying it on MEMPHIS tweets shows that LOC + TIME features provide the best results. TIME and LOC are both valuable feature groups for the classification problem compared to not using semantic abstraction. However, the results also show that using just LOD features results in a significant drop of classification performance, although, for the MEMPHIS to SEATTLE evaluation, using LOD features in combination with the other feature groups yielded the best results. This is likely to be the case because the combination of all features allows finer differentiation of LOD features even if they do not work well in isolation.

Table 6. F-Measures for training on one city and testing on a different city.

	MEMPHIS to SEATTLE	SEATTLE to MEMPHIS
ALL	**81,40% (+8,24%)**	79,07% (+7,32%)
LOC+TIME	80,43% (+7,27%)	**80,58% (+8,82%)**
LOD+TIME	55,64% (-17,52%)	71,29% (-0,46%)
LOD+LOC	69,39% (-3,78%)	74,89% (+3,14%)
LOD	64,84% (-8,32%)	64,00% (-7,75%)
LOD TYPES	64,84% (-8,32%)	63,86% (-7,89%)
LOD CATEGORIES	62,58% (-10,58%)	71,75% (0,00%)
TIME	74,72% (+1,56%)	70,43% (-1,33%)
LOC	81,13% (+7,97%)	78,22% (+6,46%)
NO Concepts	73,16% (0,00%)	71,75% (0,00%)
Majority class	49,58% (-23,59%)	53,29% (-18,46%)

Though the results are promising, we were interested to get a better understanding why the trained models work well, thus, we analyzed the rule sets in more detail. In Listing 1.4 an example rule for using all features is shown. The rule shows that location mentions in combination with incident-related keywords such as ”crash” seem to be useful as 139 true positives (TP) and no false positives (FP) are covered. The rule has a coverage of 109 TP and 3 FP in SEATTLE. Thus, it seems to be a very general rule that is universally applicable.

Listing 1.4. High-quality rule found on tweets of MEMPHIS

```
ProperLOC_TFIDF >= 0.029058, TF-IDF <= 1.433658,
    crashTFIDF >= 0.054087, clearTFIDF <= 0.139818 THEN
    Incident
```

The rule shown in Listing 1.5 is another example for a very general rule (40 TP, no FP in MEMPHIS, 294 TP in SEATTLE, 39 FP in SEATTLE). The rule contains location mentions, incident-related keywords as well as a LOD feature.

Listing 1.5. Another good rule found on tweets of MEMPHIS

```
ProperLOC_TFIDF >= 0.017093, TF-IDF <= 1.75729, carTFIDF
    <= 0, trafficTFIDF <= 0.06193, urlTFIDF <= 0.032504,
    ..//ontology/AdministrativeRegion <= 0, policeTFIDF <=
     0.080475, DDDTFIDF <= 0 THEN Incident
```

An analysis of the complete rule set shows that LOD features (5 times), temporal features (1), and location features (5) are part of the rules. Furthermore, the rule covers 20% incident-related instances in the test set compared to not using these features. All features resulting from our semantic abstraction are part of both sets, however, not surprisingly n-grams are part of the rules that are not present in the other set (12 of 14). Also the true positive rate is rather high with 85% on the test set.

A manual analysis of one rule of the model trained on MEMPHIS only using LOD features gave us a likely reason for the suboptimal performance of the classifier in SEATTLE. The rule contains the LOD features "../yago/YagoPermanentlyLocatedEntity" as well as "../yago/YagoLegalActorGeo", which have to be part of the instance more than once. For MEMPHIS this rule leads to 53 TP (no FP) whereas this rule applied on SEATTLE results in 5 TP and a total of 36 FP. Though the rules also contain several TF-IDF features and word-n-grams, a closer look at the LOD features shows that both entities which are indeed representative for incident-related tweets in MEMPHIS are indicators for not incident-related tweets in SEATTLE. This shows that LOD features cannot easily be used and need further filtering, before applying a model trained on one city on another one. A further analysis of the rule sets for just using LOD features shows that the coverage drops to 27% (-15.38%), which is an indicator that LOD features useful for MEMPHIS are indeed not useful for SEATTLE.

The analysis of the rule set for training on SEATTLE and testing on MEMPHIS shows similar results. For the ALL feature combination, LOD (5), TEMP (1), and LOC (3) features are used in the rule and all present in both datasets. In this case, all n-grams are present in the other dataset (10 of 10) that are part of the rule. Applying semantic abstraction results in an increase of coverage of the ruleset by 14% (61.75% compared to 42.38%), also increasing the true positive rate to 95% (compared to 85%).

The rule shown in Listing 1.6 is an example for a general rule of the ALL feature combination. The rule is applicable for 44 incident-related instances in the training set and applies for 91 instances in the test set without any false positives. Compared to the rule shown in Listing 1.6, similar features seem to be valuable such as TF-IDF scores and the ”crash” keyword.

Listing 1.6. A high-quality rule found on tweets of SEATTLE

```
TF-IDF <= 1.811512, crashTFIDF >= 0.057693, TF-IDF <=
    1.409797, laneTFIDF <= 0.072247 THEN Incident
```

Also in this case, just using LOD features results in a significant drop of coverage to 16.34% (-16,07%) on the test set. The rules indeed show that just one type feature is used. Also the rule set for using only categories shows that they are not part of the rules trained on SEATTLE.

Summarized, the results shown above indicate that semantic abstraction is indeed valuable for such types of classification problems. However, a combination of different feature groups seems to be necessary. Just using LOD features tends to be not valuable, due to the differences of their occurrences related to incident tweets in the two datasets.

4.3 Optimizing LOD features

As LOD features are valuable for the single-city case, but not directly for the two-city case, we manually tried to conduct a feature selection on these features. For this, we decided to use the most representative LOD features for both datasets. This resulted in eight LOD features, which are highly representative for incident-related tweets in both datasets. We confirmed our selection by merging both datasets and calculating the information gain of every single feature, leading to the ”../ontology/Road”, ”../ontology/RouteOfTransportation”, ”../ontology/ArchitecturalStructure”, and ”../ontology/Infrastructure” as the LOD features part of the top 20 features contributing the highest information gain for the combined dataset. They are also part of the eight manually selected features.

Based on this procedure, we re-evaluated the models using only these LOD features. The results presented in Table 7 show that the manual feature selection unfortunately is not valuable. This clearly indicates that more comprehensive methods for feature selection of LOD features are inevitable.

Table 7. F-measures for training on one city and testing on a different city after manual feature selection of LOD features.

	MEMPHIS to SEATTLE	SEATTLE to MEMPHIS
ALL	**81,40% (+8,24%)**	**79,07% (+7,32%)**
ALL filtered	73,08% (-0,08%)	76,88% (5,13%)
LOD	64,84% (-8,32%)	64,00% (-7,75%)
LOD filtered	66,57% (-6,60%)	63,86% (-7,89%)
NO Concepts	73,16% (0,00%)	71,75% (0,00%)

5 Related Work

Using external knowledge sources as well as information about named entities was proposed in related work several times [12], [7]. Consequently, approaches that are related to our semantic abstraction are presented. However, our approach is also related to domain adaptation, which is discussed afterwards.

Saif et al. [14] showed that adding the semantic concept for a named entity is valuable for sentiment analysis on tweets. However, their approach – extracting one concept for each named city and use it as feature – works only well for very large datasets. The authors used the concept tagging part of the AlchemyAPI to extract one concept for each named entity in a tweet and to use it as a feature. For instance, the concept ”President” is derived for ”Barack Obama”. Their results show that semantic abstraction works well for very large datasets with a multitude of topics, but not on small datasets. Compared to their work, our approach makes use of multiple types and categories extracted for a named entity, providing us with a much richer set of background information.

[4] proposed a framework for topic classification, which uses Linked Data for extracting semantic features. They compared the approach to a baseline comprising TF-IDF scores for word-unigrams, concepts extracted using the OpenCalais API, and Part-of-Speech features and showed that semantic features are indeed useful compared to the baseline approach. [17] also proposed an approach that makes use of concepts derived for instances of tweets using external knowledge databases for topic clustering. They performed a k-means clustering on tweets and showed that using conceptualized features, it is possible to outperform a plain bag-of-words approach. [20] followed a similar approach for topic clustering by using information from Wikipedia as additional features to identify topics for tweets. Also they showed an improvement compared to not using this information. [11] successfully used DBpedia resources for topic detection. Their approach is based on Part-of-Speech tagging for detecting nouns that are then interlinked to DBpedia resources using the Sem4Tags tagger.

Domain adaptation [5] also is related to our approach. However, where in domain adaptation the domains are to a large extent different, in our setting the domain, i.e., incident type classification of tweets, remains the same, the input data is subject to change. This means, that certain features, i.e., words, are changing from city to city. Therefore, feature augmentation [6] is related to our approach. However, where domain-specific features are simply discarded in regular feature augmentation, our method abstracts them in advance and then they are used in union with domain-independent features. Another way of adapting domains is structural correspondence learning [3] where shared features are identified, augmented and used to build classifiers that are applicable in both domains. The main difference is that the shared features that are then used have to be present. However, we instead create these shared features based on existing ones by the proposed semantic abstraction methods.

6 Conclusion and Future Work

In this paper we coped with the problem of generalizing a classification model in the domain of social media text classification. Using such data collected for two different cities, we were able to show that semantic abstraction is a valuable means. First, we showed that semantic abstraction indeed improves the classification of datasets derived from one city and we showed that a combination of different approaches for generating abstracted features works best (increase of F-measures by 2.14% and 1.94%, respectively). Second, semantic abstraction is also valuable when training and testing is done on two diverse datasets (increase of F-measures by 8.24% and 7.32%, respectively). However, we found that not all abstracted features contribute to a high-quality model. Especially features derived from LOD seem to be valuable for a single dataset only.

An in-depth analysis using a rule-based model showed that LOD features are indeed not directly usable for solving the generalization problem as some are representative for incident-related tweets in one dataset, but the same features are not representative on the other one. We concluded that LOD features cannot easily be used and need further filtering, before applying a model trained on one city on another one.

For future work, a first goal is to experiment with feature selection on the LOD features. However, first results using the information gain did not provide better results, thus, more sophisticated approaches are needed. As a second goal, data from more cities should be collected to get a better understanding how our approach behaves for different datasets. In this case, our first results indicate that the same findings hold true, even if different classifiers such as SVMs are used. Nevertheless, more detailed analyses are needed. Finally, additional approaches for semantic abstraction could be added such as the concept level abstraction used by [14]. We also plan to intensify our analysis of the LOD features. For instance, the relation of location mentions and incident-related tweets could be shown and was also visible in form of LOD features, however, currently we lack appropriate instruments to make use of this information.

Acknowledgements This work has been partly funded by the German Federal Ministry for Education and Research (BMBF, 01—S12054)

References

1. Agarwal, P., Vaithiyanathan, R., Sharma, S., Shroff, G.: Catching the long-tail: Extracting local news events from twitter. In: Proc. of ICWSM'12. (2012)
2. Ahn, D., van Rantwijk, J., de Rijke, M.: A cascaded machine learning approach to interpreting temporal expressions. In: Proc. NAACL-HLT, ACL (2007) 420–427
3. Blitzer, J., McDonald, R., Pereira, F.: Domain adaptation with structural correspondence learning. In: Proc. of EMNLP'06. EMNLP '06, Stroudsburg, PA, USA, Association for Computational Linguistics (2006) 120–128
4. Cano, A.E., Varga, A., Rowe, M., Ciravegna, F., He, Y.: Harnessing linked knowledge sources for topic classification in social media. In: Proc. HT '13, ACM (2013) 41–50
5. Daumé, III, H., Marcu, D.: Domain adaptation for statistical classifiers. J. Artif. Int. Res. **26**(1) (May 2006) 101–126
6. Daume III, H.: Frustratingly easy domain adaptation. In: Proc. of ACL, ACL (2007) 256–263
7. Hienert, D., Wegener, D., Paulheim, H.: Automatic classification and relationship extraction for multi-lingual and multi-granular events from wikipedia. In: Detection, Representation, and Exploitation of Events in the Semantic Web. Volume 902 of CEUR-WS. (2012) 1–10
8. Manning, C.D., Raghavan, P., Schütze., H. In: An Introduction to Information Retrieval. Cambridge University Press (2009) 117–120

9. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia Spotlight: Shedding Light on the Web of Documents. In: Proc. I-Semantics'11, ACM (2011)
10. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: Proc. of I-SEMANTICS'11, ACM (2011) 1–8
11. Muñoz García, O., García-Silva, A., Corcho, O., de la Higuera Hernández, M., Navarro, C.: Identifying topics in social media posts using dbpedia. In: Proc. of NEM Summit, Heidelberg, Germany (2011) 81–86
12. Paulheim, H.: Exploiting linked open data as background knowledge in data mining. In: Proc. of ECML/PKDD'13, DMoLD workshop. Volume 1082 of CEUR Workshop Proceedings., CEUR-WS.org (2013)
13. Ritter, A., Clark, S., Mausam, Etzioni, O.: Named entity recognition in tweets: An experimental study. In: Proc. of EMNLP '11, ACL (2011) 1524–1534
14. Saif, H., He, Y., Alani, H.: Semantic sentiment analysis of twitter. In: Proc. of ISWC'12, Springer (2012) 508–524
15. Schulz, A., Hadjakos, A., Paulheim, H., Nachtwey, J., , Mühlhäuser, M.: A multi-indicator approach for geolocalization of tweets. In: Proc. of the Seventh International Conference on Weblogs and Social Media (ICWSM). (2013)
16. Schulz, A., Ristoski, P., Paulheim, H.: I see a car crash: Real-time detection of small scale incidents in microblogs. In: Proc. of ESWC, Springer 22–33
17. Song, Y., Wang, H., Wang, Z., Li, H., Chen, W.: Short text conceptualization using a probabilistic knowledgebase. In: Proc. of IJCAI, AAAI (2011) 2330–2336
18. Strötgen, J., Gertz, M.: Multilingual and cross-domain temporal tagging. Language Resources and Evaluation (2012)
19. Witten, I.H., Frank, E.: Data mining: practical machine learning tools and techniques. Morgan Kaufman, Amsterdam, Netherlands (2005)
20. Xu, T., Oard, D.W.: Wikipedia-based topic clustering for microblogs. Proc. Am. Soc. Info. Sci. Tech. **48**(1) (2011) 1–10