Semantic Abstraction for Generalization of Tweet Classification: An Evaluation on Incident-Related Tweets

Editor(s): Guest Editors Smart Cities 2014
Solicited review(s): Alejandro Llaves, University, Country, Frank Ostermann, University, Country,
Open review(s): Fabrizio Orlandi, University, Country

Axel Schulz [a], Christian Guckelsberger [b] and Frederik Janssen [c]
[a] *Technische Universität Darmstadt, Telecooperation Lab, Germany*
E-mail: aschulz@tk.informatik.tu-darmstadt.de
[b] *Goldsmiths, University of London, Computational Creativity Group, United Kingdom*
E-mail: c.guckelsberger@gold.ac.uk
[c] *Technische Universität Darmstadt, Knowledge Engineering Group, Germany*
E-mail: 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 to process this information further. This learning problem is often concerned with regionally restricted datasets such as data from only one city. Because social media data such as tweets varies considerably across different cities, the training of efficient models requires labeling data from each city of interest, which is costly and time consuming.

To avoid such an expensive labeling procedure, a generalizable model can be trained on data from one city and then applied to data from different cities. In this paper, we present *Semantic Abstraction* to improve the generalization of tweet classification. In particular, we derive features from Linked Open Data and include location and temporal mentions. A comprehensive evaluation on twenty datasets from ten different cities shows that Semantic Abstraction is indeed a valuable means for improving generalization. We show that this not only holds for a two-class problem where incident-related tweets are separated from non-related ones but also for a four-class problem where three different incident types and a neutral class are distinguished.

To get a thorough understanding of the generalization problem itself, we closely examined rule-based models from our evaluation. We conclude that on the one hand, the quality of the model strongly depends on the class distribution. On the other hand, the rules learned on cities with an equal class distribution are in most cases much more intuitive than those induced from skewed distributions. We also found that most of the learned rules rely on the novel semantically abstracted features.

Keywords: Tweets, Classification, Linked Open Data, Semantic Abstraction, Incident Detection

1. Introduction

Social media platforms such as Twitter are widely used for sharing information about incidents. Different stakeholders, including emergency management and city administration, can highly benefit from using this up-to-date information. The large amount of new data created every day makes automatic filtering unavoidable in order to process this data further. Many approaches classify the type of an incident mentioned in social media by means of machine learning [1], [39], [40]. However, to build high-quality classifiers, labeled data is required. Given the large quantity of data in this context, creating such annotations is time consuming and therefore costly. Additionally, datasets are often naturally restricted to a certain context, i.e., labeling data from one particular city only allows training of efficient learning algorithms for exactly this city.

This is the case because the text in tweets from a particular city has special properties compared to other 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. These tokens, i.e., words, phrases or symbols, are likely to be related to the location where the text was created or contain certain topics. Thus, when the classifier relies on tokens such as named entities that are unique to the given city, e.g., street names and local sites, it is less suited for other cities where these do not occur. These aspects complicate the task of generalizing a classification model to other cities in the domain of social media texts. As an example, consider the following two tweets:

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

> "Road blocked due to traffic collision on I-495"

Both tweets comprise entities that might refer to the same thing with different wording, either on an abstract ("accident" and "car collision") or semantically rather low level ("I-90" and "I-495"). With simple syntactical text similarity approaches using standard bag of words features, it is not easily possible to make use of this semantic similarity, even though it is highly valuable for classifying both tweets.

In this paper, we introduce *Semantic Abstraction* to create a generalized model and tackle this problem. We use training information in the form of Twitter tweets that were collected in different cities. In contrast to traditional Feature Augmentation [9], our approach does not discard features prior to model creation, but makes them abstract and city-independent. In detail, we use automatic named entity and temporal expression recognition to abstract location and temporal mentions. Furthermore, we incorporate background information provided by Linked Open Data[1] (LOD) 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.

Our goal is to show that Semantic Abstraction improves classification results significantly whenever a model is trained on one city and applied on data of a different one. We therefore performed a quantitative evaluation on twenty datasets from ten different cities, involving both a two-class- and a four-class problem with more differentiated incident types. Furthermore, we conducted an in-depth analysis of the trained models. We found that (1) Semantic Abstraction indeed improves model generalization, that (2) features generated by Semantic Abstraction are frequently used and that (3) the class distribution of the training dataset affects the quality of the generalized models.

In Section 2, we present our Semantic Abstraction approach for social media data, followed by a description of our datasets in Section 3. In the following Section 4, we outline our evaluation, and present the results from two experiments: The first involves training and testing our models on data from the same cities, followed by training and testing on data from different cities. The results are then interpreted via an in-depth inspection of the learned rules. After the evaluation, we give an overview of related work in Section 5. We close with our conclusions and future work in Section 6.

2. Named Entity and Temporal Expression Recognition on Unstructured Texts

Our Semantic Abstraction approach requires us to identify named entities and expressions by means of Named Entity Recognition (NER). There is no common agreement on the definition of a named entity in the research community, and we use the following definitions throughout this paper:

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

We further define the use of toponyms in written text as a specific type of named entities:

> A **location mention** is defined as a named entity mention of a location (e.g., "Paris").

In natural language, names are not necessarily unique, e.g., there are 23 cities in the USA named "Paris", and therefore have to be disambiguated. This means that named entities may only be unique within the appropriate context. Nevertheless, in short texts

[1] http://linkeddata.org/

this contextual information is often missing [33]. However, we were able to show in prior work on tweet geolocalization [37] that the combination of different information sources helps us with coping with the disambiguation problem. In this paper, we demonstrate 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 differentiate them from named entities in the sense as they were defined before. Thus, beside NER we apply Temporal Expression Recognition and Normalization (TERN). TERN copes with detecting and interpreting temporal expressions to allow further processing. We adopt the definition of [2]:

> **Temporal expressions** are tokens or phrases in text that serve to identify time intervals (e.g., "yesterday").

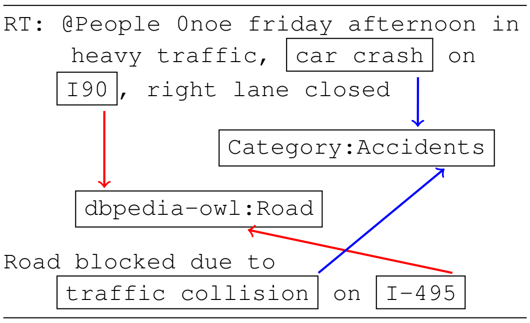
Examples for temporal expressions include "yesterday", "last Monday", "05.03.2013" and "2 hours". We apply different methods for identifying and classifying named entities and temporal expressions in tweets, and will outline them in the following subsections. Linked Open Data (LOD) is used as a source of interlinked information about various types of entities such as persons, organizations, or locations. Additionally, we apply a different NER approach for extracting location mentions and finally, we adapted a framework for the identification of temporal expressions.

2.1. *Named Entity Recognition and Replacement using Linked Open Data*

As a first approach, we use 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, e.g., "NYC", "New York City", and "The Big Apple". 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 a feature. On a semantically more abstract level, the proper location mentions "Interstate-90" and "Interstate-495" can be abstracted to the common DBpedia type "dbpedia-owl:Road". Both examples illustrate that semantic similarity between named entity mentions, or more precisely the relationship between entities, can be identified using LOD.

Listing 1 highlights two shared relations between our prior sample 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 a necessity. Third, the valuable relations have to be identified and stored. We use *DBpedia Spotlight* [25] for the first two steps in this feature generation process. In Section 3.2, we show how features are generated based on these URIs.

Listing 1: 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*

Location mentions as another type of named entity can be valuable as additional features for text classification. Linked Open Data is less useful here, because location mentions are hard to extract with DBpedia Spotlight and URIs for these entities are often missing in DBpedia. We therefore focus on the different approach of extracting *proper* location mentions as well as *common* location mentions. We found that especially the latter ones are used rather frequently in incident-related tweets, e.g., the geospatial entities "lane", "highway", or "school".

For instance, "I-90" is contained in the example tweet in Listing 1, which is a *proper* location mention. It also contains "right lane", which is a *common* location mention. With our approach, we recognize these location mentions, including different named entities such as streets, highways, landmarks, or blocks. In the following approach, both *common* and *proper* location mentions are detected and replaced with the general annotation "LOC".

For location mention extraction and replacement, we use the Stanford Named Entity Recognizer[2]. The

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

model was retrained based on 800 manually labeled tweets containing location mentions drawn from our datasets collected from Seattle, Washington and Memphis, Tennessee (see Section 3), providing more than 90% precision. The resulting model was applied to detect location mentions in both datasets for feature generation. 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 and thus considerably less generalizable but much more precise.

2.3. Temporal Expression Recognition and Replacement

Finally, we extract temporal expressions from tweets. For example, the tweet shown in Listing 1 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 [43] 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 rather informal 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 the two annotations “DATE” and “TIME”.

3. Generation of Features and Statistics of the Data

In the following, we describe how our data were collected, preprocessed, and how the features were generated. To foster a better understanding of the data, we then analyze the differences between our datasets in terms of tokens and generated LOD features.

3.1. Data Collection

We decided to focus on tweets as a suitable example for unstructured textual information shared in social media. Furthermore, we perform classification of incident-related tweets, as this type of event is highly relevant, common for all cities and not bound to a certain place. We focus both on a two-class classification problem, differentiating new tweets into "incident related" and "not incident related", and a four-class classification problem, where new tweets can be assigned to the classes "crash", "fire", "shooting", and a neutral class "not incident related". We chose these classes because we identified them as the most common incident types in Seattle using the Seattle Real Time Fire Calls data set[3], which is a frequently updated source for official incident information.

As ground truth data, we collected several city-specific datasets using the Twitter Search API[4]. These datasets were collected in a 15 km radius around the city centers of:

- Boston (USA)
- Brisbane (AUS)
- Chicago (USA)
- Dublin (IRE)
- London (UK)
- Memphis (USA)
- New York City (USA)
- San Francisco (USA)
- Seattle (USA)
- Sydney (AUS)

We selected these cities as they have a huge geographical distance, which allows us to evaluate our approaches with respect to geographical variations. Also, for all cities, sufficiently many English tweets can be retrieved. We chose 15 km as radius to collect a representative data sample even from cities with large metropolitan areas. Despite the limitations of the Twitter Search API with respect to the number of geotagged tweets, we assume that our sample is, although by definition incomplete, highly relevant to our experiments.

We collected all available tweets during certain time periods, resulting in three initial sets of tweets:

- **SET_CITY_1**: 7.5M tweets collected from November, 2012 to February, 2013 for Memphis and Seattle.
- **SET_CITY_2**: 2.5M tweets collected from January, 2014 to March, 2014 for New York City, Chicago, and San Francisco.
- **SET_CITY_3**: 5M tweets collected from July, 2014 to August, 2014 for Boston, Brisbane, Dublin, London, and Sydney.

[3] `http://seattle.data.gov`

[4] `https://dev.twitter.com/docs/api/1.1/get/search/tweets`

Table 1
Class distributions for all cities and the two classification problems.

	Two Classes		Four Classes			
	YES	NO	Crash	Fire	Shooting	No
Boston	604	2216	347	188	28	2257
Sydney	852	1991	587	189	39	2208
Brisbane	689	1898	497	164	12	1915
Chicago	214	1270	129	81	4	1270
Dublin	199	2616	131	33	21	2630
London	552	2444	283	95	29	2475
Memphis	361	721	23	30	27	721
NYC	413	1446	129	239	45	1446
SF	304	1176	161	82	61	1176
Seattle	800	1404	204	153	139	390

As manual labeling is expensive and we needed high-quality labels for our evaluation, we had to select only a small subset of these tweets. In the selection process, we first identified and gathered incident-related keywords present in our tweets. This process is described in [39]. Following this, we filtered our datasets by means of these incident-related keywords. We then removed all redundant tweets and tweets with no textual content from the resulting sets. In the next step, more than 75k tweets were manually labeled by five annotators using the CrowdFlower[5] platform. We retrieved the manual labels and selected those for which all coders agreed to at least 75%. In the case of disagreement, the tweets were removed. This resulted in twenty datasets for our evaluation, ten for each classification problem. Please note, that for the four and two-class datasets the tweets are not necessarily the same ones.[6]

Table 1 lists the detailed class distribution, and Figures 1 and 3 illustrate them in a bar chart. Overall, our datasets contain 42,808 manually labeled tweets. As the figures show, the distributions vary considerably, allowing us to evaluate our approach with typical city-specific samples. Also, the "crash" class seems to be the most prominent incident type, whereas "shootings" are less frequent. One reason for this is that "shootings" do not occur as frequently as other incidents. Another might be that people tend to report more about specific incident types and that there is not necessarily a correlation between the real-world incidents and the incidents mentioned in tweets.

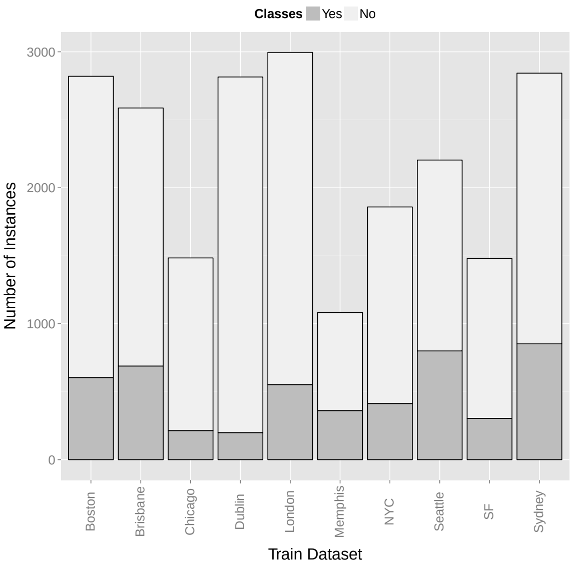


Fig. 1. Barchart of the class distributions for all cities and the two–class classification problem.

For all datasets, the class where no incident happened is the largest one. This reflects the typical situation where usually no incident occurs and only in rare cases something happens. As the following tweet shows, a reason for this is the large proportion of tweets that contain incident-related keywords but that do not refer to an actual incident:

> "I watched Air Crash Investigations on my flight from Vancouver to Toronto because apparently scaring yourself is fun."

3.2. *Preprocessing and Feature Generation*

To use our datasets for feature generation, i.e., for deriving different *feature groups* that are used for training a classification model, we had to convert the texts into a structured representation[7]. In the following we give an overview of all steps conducted for training our classification models (see Figure 2 for an overview).

Preprocessing As a first step, the text was converted to Unicode as some tweets contain non-Unicode characters, which are not processable by some preprocessing frameworks. Second, commonly used abbreviations were identified by means of a custom-made

[5] `http://www.crowdflower.com/`

[6] The datasets are publicly available at `http://www.doc.gold.ac.uk/~cguck001/IncidentTweets/`.

[7] Throughout this paper we will use *feature* as an individual property of an instance that helps training a classifier.

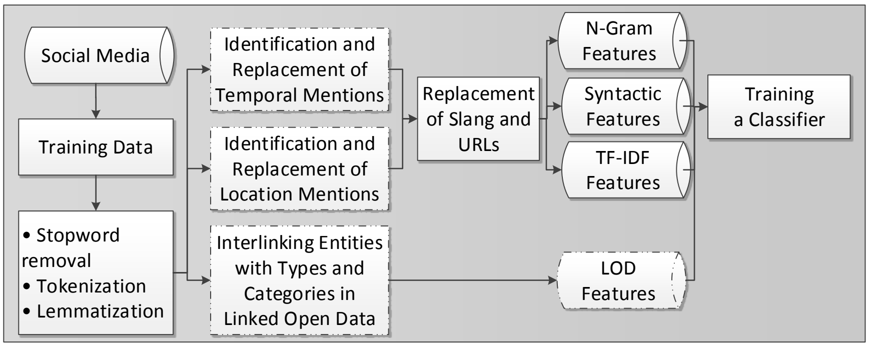

Social Media
Training Data
• Stopword removal
• Tokenization
• Lemmatization
Identification and Replacement of Temporal Mentions
Identification and Replacement of Location Mentions
Interlinking Entities with Types and Categories in Linked Open Data
Replacement of Slang and URLs
N-Gram Features
Syntactic Features
TF-IDF Features
LOD Features
Training a Classifier

Fig. 2. Semantic abstraction (in dotted lines) and feature generation steps for training a classification model for incident type classification.

dictionary based on the Internet Slang Dictionary & Translator[8]. We replaced these abbreviations with the corresponding word from formal English. Third, URLs were replaced with a common token "URL", and digits were replaced with a common token "D". We then removed stopwords and conducted tokenization on the resulting text. In this process, 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, we applied lemmatization to normalize all tokens.

Baseline Approach After this preprocessing, we extracted several features from the tweets for training. The general pipeline consists of the following steps: First, we represented tweets as a set of words. In a preliminary evaluation we found that unigrams and bigrams showed the best performance compared to longer n-grams. As features, we used a vector with the frequency of each n-gram. Second, we calculated the TF-IDF scores for each token [24]. Third, we added syntactic features such as the number of exclamation marks, questions marks, and the number of upper case characters. In the following evaluation, we treat these features as the baseline approach.

The resulting feature set was then further enhanced with our three Semantic Abstraction approaches. The approaches were used for (1) replacing already present tokens with a more general token and of (2) introducing a set of additional features derived from background knowledge about the text at hand. In contrast to the baseline approach, these approaches were performed on the original tweet, not the preprocessed one, as identifying named entities on lemmatized text is not suitable.

Semantic Abstraction using Linked Open Data As a first approach, we applied Named Entity Recognition and Replacement using Linked Open Data. To conduct this, we used the RapidMiner Linked Open Data extension [28] (the *LOD* feature group). The extension proceeds by recognizing entities based on DBpedia Spotlight [25] 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 [28], we do not treat the extracted features as binary, but use them as numeric features for our evaluation, i.e., we count how often the same feature appears in a tweet. Though it is unlikely to find the same important tokens twice in only 140 characters, different tokens such as toponyms might refer to the same URI. Counting these occurrences can help in highlighting the importance of this URI.

Finally, for each tweet, the feature encodes the number of words with the same URI. A feature selection was not conducted at this point of time, as we only have a small number of features compared to the huge number of text features in the original dataset. Also, a further analysis of token distribution of geographical close regions is open for future work.

Semantic Abstraction using Location Mentions As a second approach, 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 and 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).

[8] http://www.noslang.com/

Table 2

Percentages of overlapping tokens for two-class datasets.

	Brisbane	Boston	Chicago	Dublin	London	Memphis	NYC	SF	Seattle
Boston	19.06%								
Chicago	18.20%	18.86%							
Dublin	18.70%	19.00%	17.41%						
London	17.82%	19.52%	16.38%	19.64%					
Memphis	17.43%	17.87%	20.50%	15.28%	14.43%				
NYC	18.53%	20.12%	19.80%	18.05%	17.42%	19.03%			
San Francisco	18.48%	19.68%	19.88%	17.79%	16.95%	19.33%	19.02%		
Seattle	20.90%	22.17%	22.22%	19.69%	18.81%	23.54%	22.02%	22.26%	
Sydney	21.52%	20.42%	18.28%	20.00%	19.68%	16.83%	19.12%	18.47%	21.29%

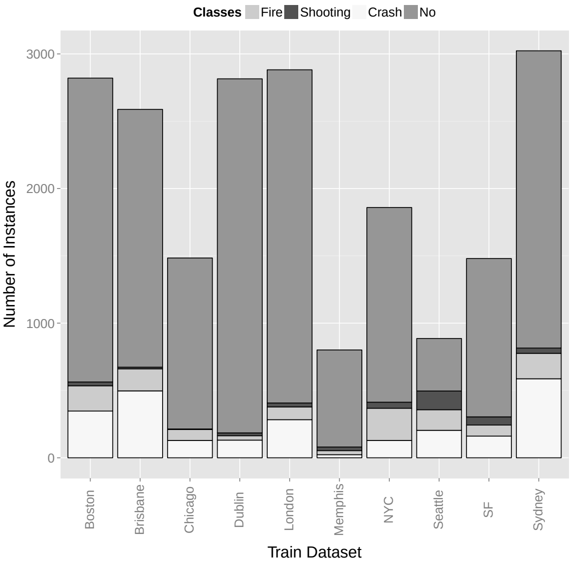


Fig. 3. Barchart of the class distributions for all cities and the four–class classification problem.

Semantic Abstraction using Temporal Expressions Third, 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 (*+TIME* feature group).

As we do not know which Semantic Abstraction approach performs best, we also provide the *+ALL* feature group in the following evaluations. This feature group is the combination of the *+LOD*, *+LOC*, and *+TIME* feature groups.

3.3. Analysis of Datasets

To foster a better understanding of the data, we analyzed the differences between our datasets in terms of tokens and generated LOD features.

Tokens One of the key questions that motivates our work is how much the used words vary in each city. We thus analyzed how similar all datasets are. Table 2 shows the percentages of overlapping unique tokens after preprocessing for the two-class datasets. The results indicate that after preprocessing, between 14% and 23% tokens are shared between the datasets. We do not assume that every unique token is a city-specific token, but the large number of tweets in our evaluations gives a first indication that there is a diversity that supports the initial hypothesis that using plain n-grams as features is not sufficient for the training of efficient models. Later on we will show that Semantic Abstraction indeed improves the training of high-quality models.

LOD Features We also analyzed the twenty most representative LOD features for the classes in all datasets. For calculating the representativeness, we counted the number of LOD features of a specific Type and Category in all two-class datasets. In Table 3 the most representative Types and Categories for the two-class datasets are shown. The selected eighteen features have considerable higher frequencies compared to all other Types and Categories.

On the one hand, the results indicate that mostly types related to location mentions are relevant for incident-related tweets. On the other hand, for both Types and Categories a large number of features is representative for the incident-related *and* the not incident-related tweets. However, also very discriminative features such as the Types *...yago/Disease114070360* (915) and *...yago/Accidents* (851) as well as the Categories *.../Category:Accidents* (1405) and *.../Category:Fire* (828) are primarily present in incident-related tweets.

Consequently, we expect that these features will also be prominent in the models. We will pick this up in our later interpretation of the rule-based models from our experiments (cf. Section 4.3).

We observed the same effects for the four-class datasets. Some of the most representative features, i.e., features that are either present for incident-related or not incident-related tweets, are shared among all four classes. However, there are also very discriminative categories for all cases such as *.../Category:Road_accidents* (825) for the crash class, *.../Category:Firefighting* (203) for the fire class, and *.../Category:Murder* (69) for the shooting class. Also in this case, mostly types related to location mentions are present for all classes. This could be an indicator that features derived for location mentions are not discriminative for the three incident classes.

In this analysis of the datasets we showed that all datasets contain a large number of unique tokens, which underlines our hypothesis that Semantic Abstraction could be useful for training a generalized model. Furthermore, the analysis also showed that LOD features only partly differentiate classes, which could be a first indication that these might not work well. In the following, we conduct an evaluation of our approaches.

4. Evaluation

We evaluated our Semantic Abstraction approaches in two critical scenarios: In the first, training and testing was performed on data from the same city. In the second, testing was performed on datasets from different cities than the ones used for training. For each scenario, we investigated both the two- and four-class problem. We want to show that Semantic Abstraction improves performance independently of the classification algorithm, and therefore evaluated each approach with five different classifiers.

The first subsection describes the methodology common to all evaluated scenarios. The sampling procedure for each scenario is described in detail in individual subsections. This is followed by both a descriptive and inferential analysis of the performances.

4.1. Method

In the following, we describe which Semantic Abstraction approaches were evaluated on which performance measure, which classifiers we used, and which statistical tests we applied. For the statistical evaluation and reporting of results, we developed and used the Open Source framework STATSREP-ML [13][9]

Semantic Abstraction Approaches We evaluated nine feature groups that emerge from the Semantic Abstraction approaches and their combinations. These feature groups extend a baseline and were described in Section 3.2. In the following sections, we reference the different combinations with their respective abbreviations: *+ALL*, *+LOC*, *+TIME*, *+LOD*, *+LOC+TIME*, *+LOC+LOD*, *+TYPES*, *+TIME+LOD*, and *+CAT*.

Classifiers In our experiments, we evaluated each feature group on a number of different classifiers. As each classifier needs to be trained several times (either in a cross validation or on data from different cities), we had to restrict their number to keep the experimental setup feasible. For training and testing, we relied on the learning algorithm implementations in the WEKA framework [14].

To ensure a fair selection, we decided to include some statistical methods such as an SVM (LibLinear) as well as symbolic ones. As often the decision trees and rule sets of tree and rule learners are quite different also in terms of performance, both types were included. Additionally, for later experiments (cf. Section 4.3), where the primary interest is what the learned models look like, it is important to have interpretable models that statistical methods are not providing.

NaiveBayes shows good performance in text classification tasks [31], which was the reason to include this algorithm. Also, the Random Forest algorithm was used as a representative of ensemble learners. We relied on the LibLinear implementation of an SVM because it has been shown that for a large number of features and a low number of instances, a linear kernel is comparable to a non-linear one [17]. Since for SVMs parameter tuning is inevitable, we evaluated the best settings for the slack variable c whenever an SVM was used. In total, we selected five classifiers, namely J48 [30], JRip [7], LibLinear [11], NaiveBayes [20], and Random Forest [4].

In summary, we have included five different classifiers that stem from the most prominent types of machine learning algorithms such as statistical or ensemble learning methods. In doing so, we want to highlight that the results of the proposed Semantic Abstraction

[9] The framework is available at `https://code.google.com/p/statistical-evaluation-for-machine-learning/`.

Table 3

Top-N incident-related types and categories with frequencies in all two-class datasets. The Not IR column indicates if the same feature is also present in not incident-related tweets.

Types			Categories		
Name	Frequency	Not IR	Name	Frequency	Not IR
...yago/YagoPermanentlyLocatedEntity	4043	x	.../Category:Article_Feedback_5_Additional_Articles	2316	x
...yago/PhysicalEntity100001930	3949	x	.../Category:HTTP	2017	x
...yago/Object100002684	3906	x	.../Category:World_Wide_Web	1993	x
...yago/YagoLegalActorGeo	3643	x	.../Category:Open_formats	1992	x
...yago/YagoGeoEntity	3464	x	.../Category:World_Wide_Web_Consortium_standards	1992	x
...ontology/Place	3374	x	.../Category:Application_layer_protocols	1992	x
...yago/Abstraction100002137	3308	x	.../Category:Web_browsers	1992	x
...ontology/PopulatedPlace	3183	x	.../Category:Road_transport	1519	
...yago/Location100027167	3126	x	.../Category:Accidents	1405	
...yago/Region108630985	3019	x	.../Category:Causes_of_death	972	
...yago/District108552138	2861		.../Category:Car_safety	892	
...ontology/Agent	2655	x	.../Category:Road_accidents	892	
...yago/AdministrativeDistrict108491826	2603		.../Category:Motorcycle_safety	851	
...yago/Whole100003553	2504		.../Category:Fire	828	
...ontology/Settlement	2423		.../Category:Road_traffic_management	810	
...ontology/MusicGenre	2143	x	.../Category:Road_safety	784	
...ontology/TopicalConcept	2143	x	.../Category:Traffic_law	780	
...ontology/Genre	2143	x	.../Category:Wheeled_vehicles	678	x

are valid in general and do not depend on a certain type of learning algorithm.

Performance Measure We used the F1-Measure for assessing performance, because it is well-established in text classification and allows one to measure the overall performance of the approaches with an emphasis on the individual classes [18]. In Section 3.3, we demonstrated that the proportion of data representing individual classes varies strongly. We therefore weighted the F1-Measure by this ratio and report the micro-averaged results.

Statistical Tests Our goal was to determine differences in the effect of different feature groups, determined by the Semantic Abstraction approaches, on classification performance. Our performance values generally do not fulfil the assumptions of normality and sphericity required by parametric tests for comparing more than two groups. It was empirically shown that under the violation of these assumptions, non-parametric tests have more power and are less prone to outliers [10]. We therefore relied exclusively on the non-parametric tests suggested in literature: Friedman's test [12,18] was used as a non-parametric alternative to a repeated-measures one-way ANOVA, and Nemenyi's test [27,18] was used post-hoc as a replacement for Tukey's test. Although it is conservative and has relatively little power, we prefer Nemenyi's test over alternatives (cf. [15]) because it is widely accepted in the machine learning community. When comparing only two groups, the Wilcoxon Signed Rank test [46,18] was used.

Unfortunately, the Friedman and Nemenyi tests only accept a single independent variable. We therefore compared the performance of individual classifiers using different feature groups, but had to average the classifier performance over the different classification algorithms to eliminate one variable. This counteracts our goal to show universal performance improvements due to Semantic Abstraction, because the information about the individual classifier performance is lost in aggregation. To support our findings, we therefore also performed tests on the F1-Measure values of the individual classifiers.

For our analysis, the feature groups used thus represent the independent variable that affects (aggregated) model performance. P-values in tables will be annotated if they are significant. While $*$ indicates low significance ($0.05 < p \leq 0.1$), the annotations $**$ and $***$ represent medium ($0.01 < p \leq 0.05$) and high significance ($p \leq 0.01$).

4.2. *Experiment 1: Same City*

In this experiment, we assessed the impact of different Semantic Abstraction approaches on the performance of classifiers that are both trained and tested on data from the same city. We therefore evaluated if Se-

Table 4

Descriptive statistics for the aggregated F1-Measurfold cross-validation with two classes.

Feature Group	Min	Max	Median	IQR
Baseline	0.855	0.956	0.907	0.027
+ALL	0.855	0.953	0.905	0.025
+LOC	0.859	0.952	0.906	0.020
+TIME	0.855	0.953	0.907	0.026
+LOD	0.851	0.953	0.905	0.023
+LOC+TIME	0.858	0.952	0.907	0.020
+LOC+LOD	0.852	0.952	0.904	0.028
+TIME+LOD	0.856	0.952	0.904	0.024
+CAT	0.857	0.952	0.906	0.027
+TYPES	0.850	0.953	0.904	0.022

Table 5

Descriptive statistics for the aggregated F1-Measurfold cross validation with four classes.

Feature Group	Min	Max	Median	IQR
Baseline	0.812	0.955	0.908	0.035
+ALL	0.788	0.954	0.904	0.031
+LOC	0.809	0.952	0.905	0.031
+TIME	0.807	0.955	0.907	0.036
+LOD	0.787	0.953	0.905	0.029
+LOC+TIME	0.817	0.953	0.906	0.031
+LOC+LOD	0.789	0.953	0.905	0.030
+TIME+LOD	0.789	0.953	0.903	0.029
+CAT	0.796	0.954	0.907	0.032
+TYPES	0.792	0.953	0.904	0.030

mantic Abstraction can support classification even if variation in tokens is low.

We performed a 10-fold cross-validation on the 20 datasets from Boston, Brisbane, Chicago, Dublin, London, Memphis, NYC, San Francisco, Seattle, and Sydney. We used stratification to ensure an equal distribution of the different classes in each fold [18]. The cross-validation was performed for every feature group and classifier algorithm, resulting in 500 raw F1-Measure values[10]. These were then reduced to 100 values by averaging over the classifiers.

Figure 6 in the appendix shows a Box-Whisker diagram of the sample distributions for the two-class problem. We also retrieved the minimum and maximum performance per sample set, as well as the median and interquartile range (IQR) as non-parametric measures for the average value and dispersion. These descriptive statistics are listed in Table 4. Similarly, Figure 7 in the appendix and Table 5 describe the sample distributions for the four class case.

For the two-class problem, the average performance varies only slightly for different feature groups, as does the persistently low dispersion. The average performance values are not much different for the four class problem, but their dispersion is a little higher. We performed a Friedman test on both sample sets. The p-values indicate weak significant differences between the performance values of the feature groups for both the two-class ($\chi_r^2(9) = 15.62, p = 0.075$) and four-class problem ($\chi_r^2(9) = 16.67, p = 0.054$). The Nemenyi test however did not show any significant pairwise differences.

Although we mainly reported the results for the aggregated performance, we will add a few results from the tests for individual classifiers to support the discussion. A comparison of the baseline results with the *+ALL* feature group using the Wilcoxon Signed-Rank test showed medium and strong significant differences in performance for the two-class case and JRip ($V = 7, p = 0.037$) and Random Forest ($V = 54, p < 0.01$). For the four-class case, we only found medium significant differences for Random Forest ($V = 47, p = 0.049$). Most notably, we also found that individual classifiers gain additional advantage by certain abstraction approaches. However, we will leave this investigation open for future work.

Discussion We did not discover any significant differences in the case of the 10-fold cross validation on a single dataset, as was suggested by the descriptive statistics. Nevertheless, the low power of the Nemenyi test and the fact that ordinary k-fold cross validations fosters type-I error inflation [10] might have complicated this investigation additionally. The results for the two-class case indicate that Semantic Abstraction is mostly not beneficial compared to the well-performing baseline. However, using the *+LOC+TIME* feature group shows a small increase for most cases. Using the *+ALL* feature group is only beneficial for the Memphis and Seattle datasets. The results for the four-class case are similar to the effects shown in the two-class case; the increase in performance whenever Semantic Abstraction is used is negligible. Solely, the *+LOC* feature group provides increases of up to 1% in F1-Measure.

Though Semantic Abstraction did not show as valuable when evaluating aggregated performance from training and testing on data from one city, we found that specific classifiers with the *+ALL* feature group significantly outperformed the equivalents that only

[10] 500 raw values = 10 feature groups × 5 classifiers × 10 folds

used the baseline features. Surprisingly, *+LOD* as the combination of the *+TYPES* and *+CAT* approaches provides significantly better results than the baseline approach. It is likely that the combination of approaches helped here in selecting more appropriate features for the classification problem. These individual comparisons show that although Semantic Abstraction does not increase general classification performance, it can be valuable for specific classifiers whenever they are trained and tested on data from the same city.

4.3. *Experiment 2: Different Cities*

In the second experiment, we trained and tested our feature groups on datasets from different cities in order to see how Semantic Abstraction is able to contribute to classification performance if the tokens in the data vary strongly. We sampled performance by means of a holdout strategy, i.e., one dataset was picked for training and the remaining nine datasets were used for testing. This was repeated ten times until all datasets were used once for testing, resulting in 4500 raw[11]- and 900 aggregated values or 90 values per feature group, respectively. We first present the results for the two-class case followed by the results for the four-class case.

The descriptive statistics for the two-class problem are listed in Table 6, and the sample distribution is illustrated in the Box-Whisker diagram in Figure 8 in the appendix. The average performance is consistently lower than in the previous scenario, and the dispersion is considerably higher. Additionally, there are larger differences between the performances. A Friedman test showed that these differences are highly significant ($\chi^2_r(9) = 190.49, p < 0.01$). We performed the Nemenyi test post-hoc to determine which feature groups differed in performance. Table 10for the pairwise comparisons.

We illustrated the ranks and significant differences between the feature groups by means of the critical distance (CD) diagram in Figure 4. Introduced by Demsar [10], this diagram lists the feature groups ordered by their rank, where lower rank numbers indicate higher performance. Two feature groups are connected if they are not significantly different. The CD diagram shows that *+ALL*, *+TIME+LOD*, *+LOC*, *+LOC+TIME*, and *+LOC+LOD* do not differ significantly. However, as shown by the Nemenyi test, these approaches differ significantly from the baseline approach, i.e., provide better performance. No statistical difference could be found for the *+TIME*, *+TYPES*, and *+CAT* approaches, which do not statistically outperform the baseline. *+LOD* as the combination of the *+TYPES* and *+CAT* approaches provides significantly better results than the baseline approach.

Table 6

Descriptive statistics for the aggregated performance from holdout sampling with two classes.

Feature Group	Min	Max	Median	IQR
Baseline	0.603	0.943	0.790	0.102
+ALL	0.653	0.937	0.817	0.073
+LOC	0.642	0.944	0.817	0.078
+TIME	0.577	0.942	0.788	0.097
+LOD	0.629	0.937	0.807	0.077
+LOC+TIME	0.649	0.946	0.816	0.087
+LOC+LOD	0.630	0.940	0.809	0.078
+TIME+LOD	0.640	0.937	0.816	0.069
+CAT	0.613	0.944	0.804	0.081
+TYPES	0.621	0.937	0.803	0.083

Table 7

Descriptive statistics for the aggregated performance from holdout sampling with four classes.

Feature Group	Min	Max	Median	IQR
+ALL	0.390	0.941	0.793	0.098
Baseline	0.359	0.941	0.775	0.108
+CAT	0.367	0.945	0.783	0.114
+LOC	0.378	0.944	0.790	0.103
+LOC+LOD	0.394	0.943	0.787	0.097
+LOC+TIME	0.413	0.944	0.790	0.107
+LOD	0.387	0.945	0.789	0.095
+TIME	0.342	0.942	0.774	0.114
+TIME+LOD	0.389	0.944	0.793	0.097
+TYPES	0.371	0.942	0.786	0.095

Table 7 shows the descriptive statistics for the four-class problem, and Figure 9 in the appendix illustrates the sample distributions. The average performance is in many cases lower than for the two-class problem, and the dispersion is a bit higher for most feature groups. The Friedman test again shows strong significant differences between the performances of the feature groups ($\chi^2_r(9) = 165.02, p < 0.01$). The Nemenyi test indicates strong significant pairwise differences and the p-values are listed in Table 11 We again use a critical distance diagram (see Figure 5) to highlight the significant pairwise differences. It shows that *+ALL*, *+TIME+LOD*, *+LOD*, *+LOC*, *+LOC+TIME*, *+LOC+LOD*, and *+TYPES* do not differ significantly

[11] 4500 raw values = 10 feature groups × 5 classifiers × 9 testing datasets × 10 datasets

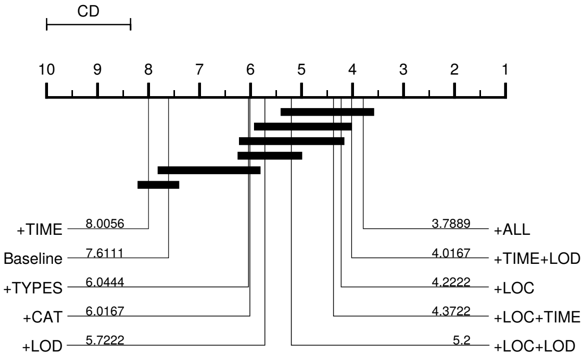


Fig. 4. Critical distance (CD) diagram showing a comparison of all feature groups against each other based on the holdout sampling with two classes. Feature groups that are not significantly different at $p = 0.01$ are connected.

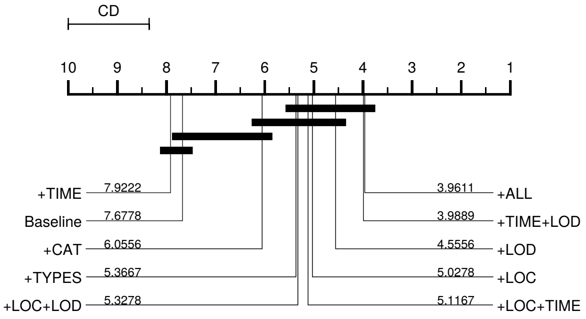


Fig. 5. Critical distance (CD) diagram showing a comparison of all feature groups against each other based on the holdout sampling with four classes. Feature groups that are not significantly different at $p = 0.01$ are connected.

with respect to their performance, but outperform the baseline. As in the two-class case, *+TIME* and *+CAT* do not outperform the baseline.

As in the first experiment, we want to support these findings with comparisons of the baseline against the *+ALL* feature group for individual classifiers. For the 2-class problem, the Wilcoxon Signed-Rank test indicated medium and strong significant differences for all classifiers, in detail for J48 ($V = 1477, p = 0.022$), JRip ($V = 1235, p < 0.01$), LibLinear ($V = 1330, p < 0.01$), Naive Bayes ($V = 350.5, p < 0.01$) and Random Forest ($V = 538, p < 0.01$). For the four-class case, low significant differences were found for J48 ($V = 1588, p = 0.065$), and strong significance for JRip ($V = 1388, p < 0.01$), LibLinear ($V = 1183, p < 0.01$), Naive Bayes ($V = 248.5, p < 0.01$), and Random Forest ($V = 821, p < 0.01$). These results show that all individual classifiers can indeed benefit from our Semantic Abstraction.

Discussion The results support our hypothesis that Semantic Abstraction can contribute significantly to classification performance when training and testing is not performed on data from the same city.

As we wanted to get a better understanding of the effects caused by Semantic Abstraction, we investigated two crucial questions regarding the properties of the different datasets:

1. How well does a model that was trained on datasets of other cities perform on the dataset of the city at hand?
2. What makes a good training set, i.e., how well does the dataset of a particular city at hand serve as a training set to build a classifier for datasets from other cities?

For answering these questions, we investigated the individual classification performance for datasets from each city. Also, to backup our findings, we examined the rules learned by the JRip rule learner in more detail. The effects of Semantic Abstraction on the models can best be grasped by considering the rule sets with a focus on the semantic features *+LOC*, *+TIME*, and *+LOD*.

Individual Classification Performance We first examine the average F1-Measure for datasets from each city in more detail. Table 8mance values whenever the *+ALL* feature groups are used for all cities for the two-class problem. Table 9 shows the results for theted the *+ALL* model as it showed the best performance (cf. Figures 4 and 5). In the two-class problem, the datasets for the two cities Dublin and Chicago (depicted in bold in Table 8 at the left hand side) have shown the best classification performance (about 2-3% F1-Measure less thanowever, the datasets for Seattle and Sydney performed poorly with up to 15% less. As we show, the reason for the good performance of the Dublin and Chicago datasets is strongly related to the class distribution. For the Dublin dataset, the baseline F1-Measure is the highest while the Chicago dataset ranks at the second place[12]. Interestingly, the same holds for the datasets of the two cities that are rather hard to classify. These two are ranked at or near the bottom in terms of baseline F1-Measure (Seattle being the worst and Sydney the third worst).

[12] The baseline F1-Measure is computed by utilizing a classifier that predicts the *majority class*, i.e., the class with the most tweets, for all instances.

Table 8

Two-class problem: F1-Measure for training on data of one city (header) using the *+ALL* feature groups and applying on a different city (first column). Also, the datasets of the cities that were best (**bold**) and worst (*emphasis*) to classify (in first column) and best (bold) and worst (emphasis) to train on (header) are shown.

	Boston	Brisbane	*Chicago*	*Dublin*	London	Memphis	NYC	SF	**Seattle**	Sydney	Average F1-Measure	Difference to Single-city Model	Majority Class
Boston		86.00%	83.99%	79.73%	84.81%	81.75%	84.50%	84.56%	83.41%	87.15%	83.99%	-7.56%	78.58%
Brisbane	86.60%		82.75%	77.49%	79.90%	77.22%	80.61%	82.42%	84.16%	84.32%	81.72%	-11.05%	73.37%
Chicago	91.40%	89.26%		86.51%	88.35%	87.08%	88.72%	89.27%	87.80%	89.28%	88.63%	-3.43%	85.58%
Dublin	93.29%	92.78%	93.17%		93.67%	91.05%	92.54%	92.96%	92.06%	93.00%	92.72%	-2.74%	92.93%
London	81.06%	79.58%	77.95%	80.87%		78.31%	78.89%	79.50%	79.74%	81.10%	79.67%	-10.85%	81.58%
Memphis	80.92%	80.71%	77.22%	68.58%	74.57%		75.77%	80.02%	84.12%	75.76%	77.52%	-8.56%	66.63%
NYC	85.52%	82.56%	80.17%	75.21%	79.11%	78.50%		81.28%	78.52%	80.65%	80.17%	-9.6%	77.78%
San Francisco	86.29%	84.73%	83.42%	80.49%	82.17%	81.98%	83.10%		82.91%	83.63%	83.19%	-6.97%	79.45%
Seattle	74.48%	75.57%	69.28%	65.26%	71.65%	73.48%	71.77%	71.51%		74.46%	71.94%	-13.54%	63.70%
Sydney	82.68%	83.48%	74.04%	72.30%	79.53%	72.97%	75.08%	77.70%	80.50%		77.59%	-15.61%	70.03%
Avg. F1-Measure	84.69%	83.58%	79.75%	75.84%	81.12%	80.07%	80.81%	81.83%	83.73%	82.77%			

To support this observation, we computed the Spearman Rank Correlation [23] between two rankings: first, we calculated the differences in F1-Measure between the cross validation result using the baseline approach, i.e., the baseline single-city model, and the average of all train/test splits using the *+ALL* approach. The training datasets were then ranked according to the differences. Second, we determined the majority class of each dataset to have a representation of the actual class distribution and also ranked the datasets accordingly.

The analysis of the rankings using Spearman's rho for the two-class problem indicates that the ranking of the differences and the majority classes are significantly positively correlated ($\rho(10) = 0.758, p < 0.05$, two-tailed test). We thus can conclude that classification performance is related to the class distribution, i.e., better classification performance is correlated with a larger majority class. Please keep in mind that we trained on several datasets with different distributions, thus, this observation is not trivial and not comparable to a single-city case.

Interestingly, the situation appears to be similar when four classes have to be predicted (cf. Table 9t works best with only 3% less F1-Measure but the Chicago dataset was ranked third as the Memphis dataset showed to be very well classified. The Seattle dataset performed worst with a crude 34% lower F1-Measure, followed by the Sydney dataset as in the two-class problem (both shown in italics in Table 8). The analysis of these rho indicates that the ranking of the differences and the majority classes are strong significantly positively correlated ($\rho(10) = 0.988, p < 0.01$, two-tailed test). We thus can also conclude that a better classification performance for the four-class problem is correlated with a larger majority class.

Thus, datasets of cities with a skewed class distribution, i.e., with a larger majority class, can better be classified compared to more equally distributed classes. This holds true for both the two-class problem and the four-class one. However, for the latter the distribution is even more important as the dataset with the worst performance showed a 33% performance drop compared to the model learned just using data of that city.

Quality of Training Set In the following, we cope with the second question, i.e., which datasets serve well to create a training set on. Answering this question could contribute to a better selection of cities to train models for other cities on. Also, the models themselves are of interest as on the one hand the improvement Semantic Abstraction brings can be implicitly shown and on the other hand valuable features for the classifier can be identified. For this investigation it is crucial to understand the models that have been trained, thus, we rely on the rules learned by the JRip algorithm and proceed with a similar procedure as in [19]. Please note that different feature groups might be used for different classifiers.

For the two-class classification problem, the two datasets for Boston and Seattle yielded the highest performance with about 85% and 84% (depicted in bold in the header of Table 8 and Dublin showed the worst results with only about 76% and 80%. Dublin and Chicago were best to classify (cf. Table 9) but now are worst to leaasizes the importance of an appropriate class distribution and at least partly confirms prior findings that more equally distributed classes often yield more robust classifiers [44,45]. Not surpris-

Table 9

Four-class problem: F1-Measure for training on data of one city (header) using the *+ALL* feature groups and applying on a different city (first column). Also, the datasets of the cities that were best (**bold**) and worst (*emphasis*) to classify (in first column) and best (bold) and worst (emphasis) to train on (header) are shown.

	Boston	Brisbane	Chicago	*Dublin*	London	*Memphis*	NYC	SF	Seattle	**Sydney**	Average F1-Measure	Difference to Single-city Model	Majority Class
Boston		85.29%	83.93%	79.64%	84.26%	73.11%	83.35%	83.67%	81.17%	86.76%	82.35%	-9.68%	80.03%
Brisbane	84.81%		81.97%	74.91%	80.12%	68.06%	76.84%	81.99%	78.17%	84.92%	79.09%	-13.6%	73.99%
Chicago	90.37%	88.93%		84.97%	87.83%	79.70%	87.45%	87.53%	83.39%	88.51%	86.52%	-5.63%	85.58%
Dublin	93.66%	93.25%	93.28%		94.12%	89.04%	93.09%	93.42%	88.79%	93.77%	92.49%	-2.98%	93.43%
London	89.69%	89.32%	88.13%	89.09%		78.66%	87.68%	88.07%	83.10%	90.14%	87.1%	-6.34%	85.88%
Memphis	86.54%	85.89%	85.87%	85.56%	86.02%		86.34%	86.24%	86.49%	85.10%	86.01%	-3.28%	90.00%
NYC	79.85%	79.20%	78.56%	73.28%	77.05%	68.89%		78.25%	77.36%	78.58%	76.78%	-12.15%	77.78%
San Francisco	84.19%	81.80%	81.37%	78.36%	81.01%	72.32%	79.41%		80.14%	81.08%	79.97%	-9.93%	79.45%
Seattle	51.79%	51.01%	42.70%	39.05%	45.00%	49.27%	45.16%	50.48%		53.21%	47.52%	-33.68%	44.01%
Sydney	81.11%	80.36%	74.42%	74.46%	76.38%	62.69%	72.26%	75.38%	75.70%		74.75%	-18.27%	73.04%
Avg. F1-Measure	82.45%	81.22%	78.29%	74.96%	78.44%	71.08%	78.53%	80.17%	81.64%	81.91%			

ingly, the rules learned on data from these two cities with skewed class distributions also showed to be suboptimal.

$$incident \leftarrow \text{dbpedia.org/Category:Accidents} \geq 1, \text{dbpedia.org/ontology/MusicGenre} \geq 1 \quad (1)$$

For instance, the first rule learned on the dataset of Dublin (cf. Rule 1) tests for the category accidents and for the music genre, which clearly is a bad choice.

$$incident \leftarrow \text{dbpedia.org/Category:Trucks} \geq 1, \text{url} \geq 1 \quad (2)$$

Another rule tests for the category trucks combined with a check for the presence of a URL (cf. Rule 2. The first condition might be plausible for the dataset at hand, but in general trucks are not involved more often in incidents than other vehicles[13]. Thus, this rule seems to be valid for Dublin only and yields suboptimal classification results in other cities.

$$incident \leftarrow \text{dbpedia.org/Category:Accidents} \geq 1 \quad (3)$$

In Chicago, the first rule checks for the category accidents which seems to be a good choice (the rule covers 82 positives and 28 negatives, cf. Rule 3). But already the third rule tests if the word “woman” is present, which is not a good indicator for an incident related tweet (see Rule 4). Despite, it covers 22 positive examples and 5 negative ones which means that the rule works well in Chicago. Nevertheless, in other cities, “women” are usually not directly related to incidents.

$$incident \leftarrow \text{woman} \geq 1 \quad (4)$$

The other way around, the rules for the datasets of Boston and Seattle show good generalization capabilities. For example, the first rule for the Boston dataset checks the category accidents and if at least six letters are upper case (cf. Rule 5). Usually, upper case letters are an indicator that people want to emphasize the content of a tweet, e.g., during incidents.

$$incident \leftarrow \text{dbpedia.org/Category:Accidents}, \text{UPPER\_CASE} \geq 6 \quad (5)$$

The second one tests for the word “fire” in addition to the +LOD feature “city” (Rule 6) and the third rule simply considers the word “accident” (Rule 7). In the Seattle dataset, the first six rules all contain abstracted location mentions in their body in conjunction with different other features. The first one additionally checks for the word “lane”, the second the TF-IDF of the word “unit” (i.e., a specific police squad) and the

[13] Table 1107 of census data from 2009 shows that actually a lot less trucks were involved in crashes compared to cars or other vehicles, cf. http://www.census.gov/compendia/statab/2012/tables/12s1107.pdf, accessed 2015-02-02

third the TF-IDF for “crash”. All of these rules appear to be also universally valid.

$$incident \leftarrow \text{fire} \geq 1, \text{dbpedia.org/ontology/City} \quad (6)$$

$$incident \leftarrow \text{accident} \geq 1 \quad (7)$$

For the four-class problem, the datasets of Boston and Sydney worked best to train a model on while the datasets of Dublin and Memphis performed worst when used for training. Again, these two showed the best performance in classification. About 30% of the rules for the Boston dataset had +LOD features in their body where some were plausible (such as *category:Security* (class “shooting”), *category:Fire* (class “fire”), *category:Accidents* for the class “crash”) but, however, there were also some rather unintuitive ones such as specific geographic places or *Early_American_Industrial_Centers* (class “fire”). In the Sydney dataset, more exotic categories like *Wildland_Fire_Suppression* or *Former_Member_States_Of_The_United_Nations* are present for the class “fire”. However, also here some rather unintuitive features were included in the rules.

In summary, the class distribution to a high extent affects the quality of the models. This is not only reflected by the correlation between the difference of the models of datasets from other cities to that induced on datasets of the same city, but also by the inspected rule sets. Here, we showed that in many cases the rules found on datasets of the cities that are better suited to train a classifier on are also more intuitive than those of the cities where no high-quality model could be learned. Also, we showed that Semantic Abstraction indeed is valuable to build high-quality models as most often the novel semantically abstracted features were used in the rules.

5. Related Work

In the domain of machine learning, there already exists related work covering the use of external knowledge sources as well as information about named entities [28], [16]. In this section, we will describe the methods related to our Semantic Abstraction approach in more detail. However, our approach is also related to the field of domain adaptation, which is discussed afterwards. Finally, we present related approaches in the domain of incident type classification.

Saif et al. [35] showed that adding the semantic concept for a named entity is valuable for sentiment analysis on tweets. The authors used the concept tagging component of the AlchemyAPI to extract one concept for each named entity in a tweet, and then used it as a feature. For instance, the concept “President” is derived from “Barack Obama”. This approach 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.

Cano et al. [5] 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[14], and Part-of-Speech features and showed in comparison with this baseline that semantic features can indeed support topic classification. Song et al. [41] also proposed an approach that makes use of concepts derived from tweets using external knowledge databases for topic clustering. They performed a k-means clustering on tweets and demonstrated that using conceptualized features, it is possible to outperform a plain bag-of-words approach. Xu and Oarg [47] followed a similar approach for topic clustering by using information from Wikipedia as additional features to identify topics for tweets. They were also able to show that this information improves clustering. Muñoz et al. [26] successfully used DBpedia resources for topic detection. Their approach is based on Part-of-Speech tagging for detecting nouns that are then linked to DBpedia resources using the Sem4Tags tagger[15].

Domain adaptation [8] 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, but the input data is subject to change. This means, that certain features, i.e., tokens, are changing from city to city. Therefore, feature augmentation [9] 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]

[14] `http://www.opencalais.com`

[15] `http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/technologies/138-sem4tags`

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.

Several prior work focused on the classification of incident-related user-generated content. The works of [36], [6], [34], and [21] were designed for incident type classification. However, the models were trained on traditional features such as word-n-grams. Also, Twitter-specific features such as hashtags and @-mentions have been used. In contrast to these approaches, several works tried to incorporate information about named entities present in tweets, which is somehow related to our Semantic Abstraction approach. Agarwal et al. [1] proposed an approach for classifying tweets related to a fire in a factory. As features, they use the number of occurrences of certain named entities such as locations, organizations, or persons that are extracted using the Stanford NER toolkit. Furthermore, the occurrence of numbers and URLs is used as a feature. Also, word occurrences remaining after stopword filtering are used. Similarly, Li et al. [22] built a classifier using text features and Twitter-specific features, such as hashtags, @-mentions, URLs, and the number of spatial and temporal mentions for incident type classification. Nevertheless, the authors do not specifically try to generalize the tokens at hand, but only introduce a set of features based on identified named entities. In [42] clusters were built in order to identify huge forest fires. However, the found clusters had to be inspected manually and no supervised learning method was used. In summary, none of the prior works focus on datasets of different cities as we did in our evaluation.

6. Conclusion and Future Work

In this paper, we introduced Semantic Abstraction to foster the generalization of classification models in the domain of social media text classification. Using Twitter data collected from ten different cities, we were able to show that our approach is indeed very useful, especially in the domain of incident type classification.

We first demonstrated that Semantic Abstraction can also improve the classification of datasets derived from only one city. Nevertheless, we also found that Semantic Abstraction is most valuable when training and testing is done on datasets from different cities, i.e., with a diverse range of tokens. Furthermore, we discovered that the success of our approach in this scenario depends on the choice of the classifier. We noticed that not all feature groups and abstracted features contribute to a high-quality model. Especially features derived using the temporal expression recognition approach and features based on LOD seem to need further rework. Finally, an in-depth analysis of the train/test case showed that the class distribution affects the quality of the models to a high extent. This is an important finding, as it might help to create datasets that generalize better.

Nevertheless, training such a generalizing classification model is highly desirable for several reasons. For instance, generalizing models only have to be trained once, and can therefore help decreasing time and costs of training data labelling. Furthermore, they can help to ensure high classification rates even in areas where only few training samples are available. Finally, in times of increasing growth of cities and the merging with surrounding towns to large metropolitan areas, they allow to cope with the latent transitions in token use.

The applicability of Semantic Abstraction for a real-world use case has also been shown in our prior work [40]. In that paper, we proposed an approach for small-scale incident detection based on spatial-temporal-type clustering. We employed Semantic Abstract and extracted generalized features to increase robustness a classification model that identifies incident-related tweets.

For future work, a first goal is to experiment with feature selection on the LOD features. Because preliminary experiments using the information gain did not indicate better results [38], more sophisticated approaches such as the one presented by Ristoski and Paulheim [32] will be needed. A second goal is to include additional approaches for Semantic Abstraction such as the concept of level abstraction used by Saif et al. [35]. We also plan to intensify our analysis of the LOD features. For instance, we pointed out the relation between location mentions and incident-related tweets, which is also visible in form of LOD features. However, currently we lack appropriate instruments to make use of this information.

Also, a more diverse set of Linked Data sources might contribute to our approach. However, as our NER approach is based on DBpedia Spotlight, it might be difficult to relate named entities to the corresponding URIs of other sources. A solution might be a brute force approach as proposed by [29], that creates and matches URI patterns based on the named entity mentions. Furthermore, SPARQL endpoints or REST APIs might also be used for efficient retrieval of URIs. Nevertheless, in all of these cases, one needs to take care of the disambiguation steps.

Additionally, also non-LOD sources could improve Semantic Abstraction. Spatial data such as the information provided by GeoNames [16], which was used in our previous work [37], might provide additional information about locations. Furthermore, company or governmental data, including domain-specific information, could also be used as additional features.

In a future analysis, we plan to investigate the relation between specific tokens and the Semantic Abstraction in more detail for individual cities. The goal is to identify which approach works best given a certain type of token. Also, in order to increase the general validity of our results, we plan to include more classifiers in the experiments.

Acknowledgments

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

References

[1] P. Agarwal, R. Vaithiyanathan, S. Sharma, and G. Shroff. Catching the long-tail: Extracting local news events from twitter. In J. G. Breslin, N. B. Ellison, J. G. Shanahan, and Z. Tufekci, editors, *Proceedings of the Sixth International Conference on Weblogs and Social Media (ICWSM'12), Dublin, Ireland*. The AAAI Press, 2012.

[2] D. Ahn, J. van Rantwijk, and M. de Rijke. A cascaded machine learning approach to interpreting temporal expressions. In L. Vanderwende, H. D. III, and K. Kirchhoff, editors, *Proceedings of the The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT'07), Rochester, NY*, pages 420–427. The Association for Computational Linguistics, 2007.

[3] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In D. Jurafsky and Éric Gaussier, editors, *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP'06), Sydney, Australia*, EMNLP '06, pages 120–128, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[4] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[5] A. E. Cano, A. Varga, M. Rowe, F. Ciravegna, and Y. He. Harnessing linked knowledge sources for topic classification in social media. In G. Stumme and A. Hotho, editors, *Proceedings of the 24th ACM Conference on Hypertext and Social Media (HT'13), Paris, France*, pages 41–50, New York, NY, USA, 2013. ACM.

[6] L. S. Carvalho, S. and R. Rossetti. Real-time sensing of traffic information in twitter messages. In *Proceedings of the 4th Workshop on Artificial Transportation Systems and Simulation (ATSS'10), Madeira, Portugal*, pages 19–22. IEEE Computer Society, 2010.

[7] W. W. Cohen. Fast Effective Rule Induction. In A. Prieditis and S. J. Russell, editors, *Proceedings of the 12th International Conference on Machine Learning (ICML'95), Tahoe City, California, USA*, pages 115–123. Morgan Kaufmann, 1995.

[8] H. Daumé, III and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research (JAIR)*, 26(1):101–126, May 2006.

[9] H. Daume III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL'07)*, pages 256–263, Prague, Czech Republic, 2007. Association for Computational Linguistics.

[10] J. Demšar. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, Jan(7):1–30, 2006.

[11] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[12] M. Friedman. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92, 1940.

[13] C. Guckelsberger and A. Schulz. STATSREP-ML: Statistical evaluation & reporting framework for machine learning results. Technical report, TU Darmstadt, tuprints, 2015.

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

[15] F. Herrera. An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons. *Journal of Machine Learning Research*, 9:2677–2694, 2008.

[16] D. Hienert, D. Wegener, and H. Paulheim. 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 (DeRiVE'12)*, volume 902 of *CEUR-WS*, pages 1–10, 2012.

[17] C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University, 2003.

[18] N. Jakowicz and M. Shah. *Evaluating Learning Algorithms. A Classification Perspective*. Cambridge University Press, Cambridge, 2011.

[19] F. Janssen. *Heuristic Rule Learning*. PhD thesis, TU Darmstadt, Knowledge Engineering Group, Darmstadt, Germany, Oct. 2012.

[20] G. H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI'95), Montreal, Canada*, pages 338–345, San Francisco, CA, USA, 1995. Morgan Kaufmann.

[21] S. Karimi, J. Yin, and C. Paris. Classifying microblogs for disasters. In *Proceedings of the 18th Australasian Document Computing Symposium (ADCS'13), Brisbane, Queensland, Australia*, pages 26–33, New York, NY, USA, 2013. ACM.

[16] `http://www.geonames.org`

[22] R. Li, K. H. Lei, R. Khadiwala, and K. C.-C. Chang. Tedas: A twitter-based event detection and analysis system. In A. Kementsietsidis and M. A. V. Salles, editors, *Proceedings of the 28th International Conference on Data Engineering (ICDE'12), Washington, DC, USA*, pages 1273–1276. IEEE Computer Society, 2012.

[23] A. D. Lovie. Who discovered spearman's rank correlation? *British Journal of Mathematical and Statistical Psychology*, 48(2):255–269, 1995.

[24] C. D. Manning, P. Raghavan, and H. Schütze. *An Introduction to Information Retrieval*, pages 117–120. Cambridge University Press, 2009.

[25] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. DBpedia Spotlight: Shedding Light on the Web of Documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics'11), Graz, Austria*, pages 1–8, New York, NY, USA, 2011. ACM.

[26] O. Muñoz García, A. García-Silva, O. Corcho, M. de la Higuera Hernández, and C. Navarro. Identifying topics in social media posts using dbpedia. In *Proceedings of NEM Summit*, pages 81–86, Heidelberg, Germany, 2011.

[27] P. B. Nemenyi. *Distribution-free Multiple Comparisons*. PhD thesis, Princeton University, 1963.

[28] H. Paulheim. Exploiting linked open data as background knowledge in data mining. In C. d'Amato, P. Berka, V. Svátek, and K. Wecel, editors, *Proceedings of the International Workshop on Data Mining on Linked Data, with Linked Data Mining Challenge collocated with the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (DMoLD'13), Prague, Czech Republic*, volume 1082 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.

[29] H. Paulheim and J. Fürnkranz. Unsupervised Feature Generation from Linked Open Data. In *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics (WIMS'12), Craiova, Romania*, pages 31:1–31:12, New York, NY, USA, 2012. ACM.

[30] R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[31] J. D. Rennie, L. Shih, J. Teevan, and D. R. Karger. Tackling the poor assumptions of naive bayes text classifiers. In T. Fawcett and N. Mishra, editors, *Proceedings of the International Conference on Machine Learning (ICML-03)*, pages 616–623. AAAI Press, 2003.

[32] P. Ristoski and H. Paulheim. Feature selection in hierarchical feature spaces. In *Proceedings of the 17th International Conference on Discovery Science (DS'14), Bled, Slovenia*, volume 8777 of *Lecture Notes in Computer Science*, pages 288–300. Springer, 2014.

[33] A. Ritter, S. Clark, Mausam, and O. Etzioni. Named entity recognition in tweets: An experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*, pages 1524–1534, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[34] D. R. Robert Power, Bella Robinson. Finding fires with twitter. In *Proceedings of the Australasian Language Technology Association Workshop (ALTA'13), Brisbane, Australia*, pages 80–89. Association for Computational Linguistics, 2013.

[35] H. Saif, Y. He, and H. Alani. Semantic sentiment analysis of twitter. In *Proceedings of the 11th International Conference on The Semantic Web - Volume Part I (ISWC'12), Boston, MA*, pages 508–524, Berlin, Heidelberg, 2012. Springer.

[36] T. Sakaki and M. Okazaki. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World Wide Web (WWW'10), Raleigh, North Carolina, USA*, pages 851–860, New York, NY, USA, 2010. ACM.

[37] A. Schulz, A. Hadjakos, H. Paulheim, J. Nachtwey, , and M. Mühlhäuser. A multi-indicator approach for geolocalization of tweets. In E. Kiciman, N. B. Ellison, B. Hogan, P. Resnick, and I. Soboroff, editors, *Proceedings of the Seventh International Conference on Weblogs and Social Media (ICWSM'13)*. The AAAI Press, 2013.

[38] A. Schulz and F. Janssen. What is good for one city may not be good for another one: Evaluating generalization for tweet classification based on semantic abstraction. In T. Omitola, J. Breslin, and P. Barnaghi, editors, *Proceedings of the Fifth Workshop on Semantics for Smarter Cities a Workshop at the 13th International Semantic Web Conference, Riva del Garda, Italy*, volume 1280, pages 53–67. CEUR, 2014.

[39] A. Schulz, P. Ristoski, and H. Paulheim. I see a car crash: Real-time detection of small scale incidents in microblogs. In *The Semantic Web: ESWC 2013 Satellite Events*, volume 7955 of *Lecture Notes in Computer Science*, pages 22–33. Springer-Verlag, 2013.

[40] A. Schulz, B. Schmidt, and T. Strufe. Small-scale incident detection based on microposts. In *Proceedings of the XXX Conference on Hypertext*. ACM, 2015.

[41] Y. Song, H. Wang, Z. Wang, H. Li, and W. Chen. Short text conceptualization using a probabilistic knowledgebase. In T. Walsh, editor, *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three (IJCAI'11), Barcelona, Catalonia, Spain*, pages 2330–2336. AAAI Press, 2011.

[42] L. Spinsanti and F. Ostermann. Automated geographic context analysis for volunteered information. *Applied Geography*, 43:36–44, Sept. 2013.

[43] J. Strötgen and M. Gertz. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47:269–298, 2012.

[44] G. M. Weiss. *The Effect of Small Disjuncts and Class Distribution on Decision Tree Learning*. PhD thesis, Department of Computer Science, Rutgers University, New Brunswick, New Jersey, May 2003.

[45] G. M. Weiss and F. Provost. Learning when training data are costly: The effect of class distribution on tree induction. *Journal of Artificial Intelligence Research*, 19(1):315–354, Oct. 2003.

[46] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1(6):80–83, 1945.

[47] T. Xu and D. W. Oard. Wikipedia-based topic clustering for microblogs. *Proceedings of the Association for Information Science and Technology*, 48(1):1–10, 2011.

Appendix

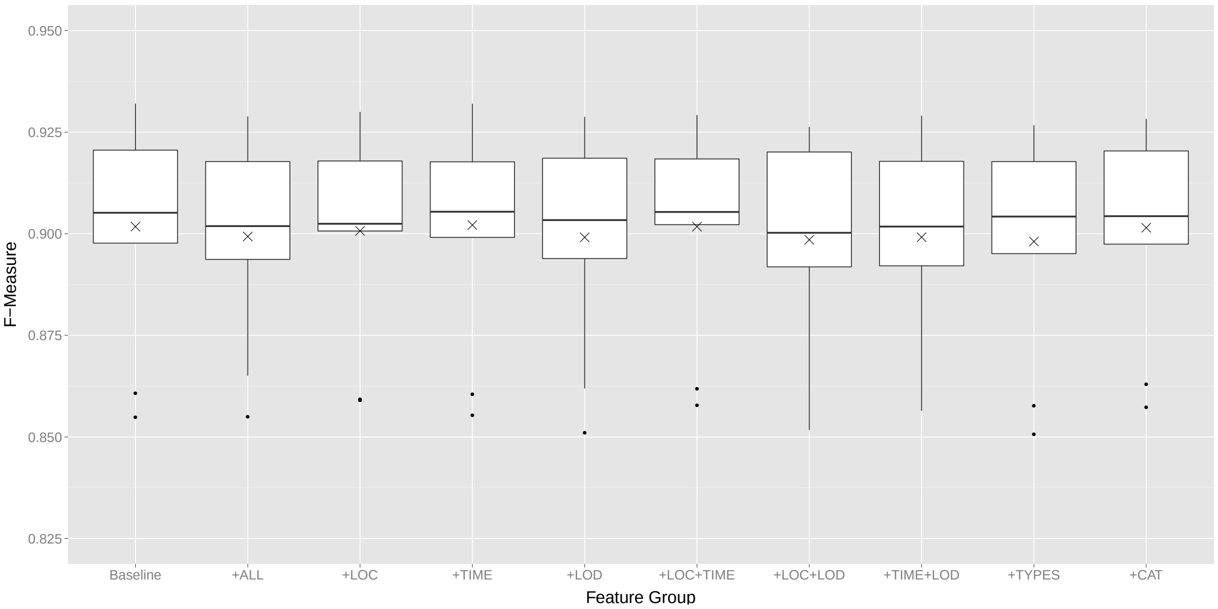


Fig. 6. Box-Whisker diagram for the aggregated performance from 10-fold cross-validation with two classes. The mean is indicated by a cross.

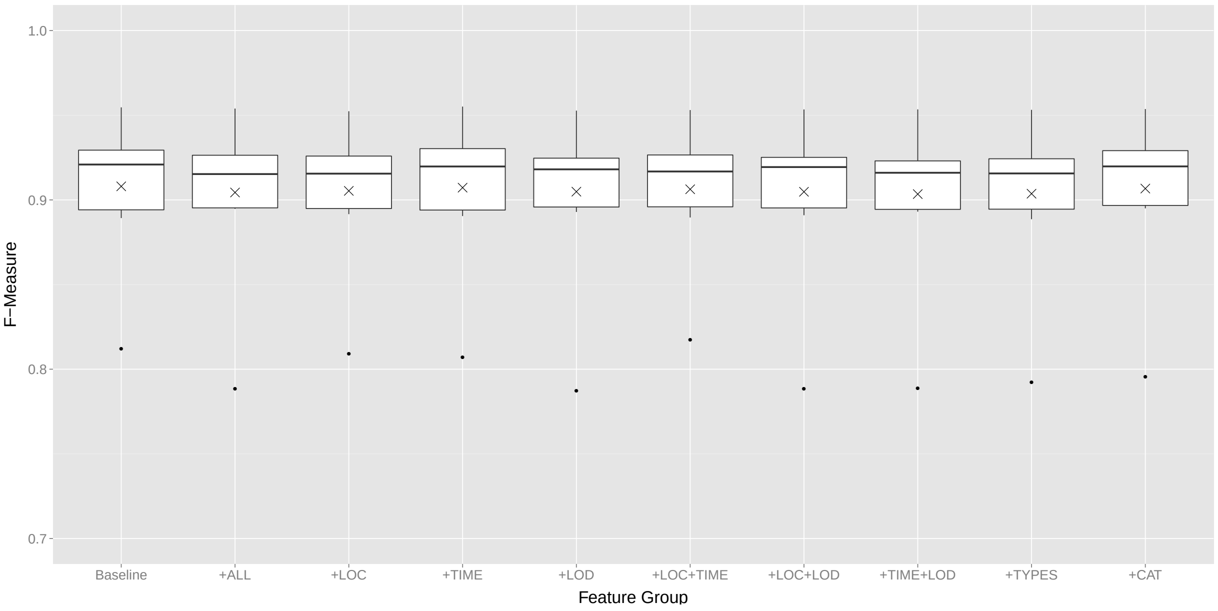


Fig. 7. Box-Whisker diagram for the aggregated performance from 10-fold cross-validation with four classes. The mean is indicated by a cross.

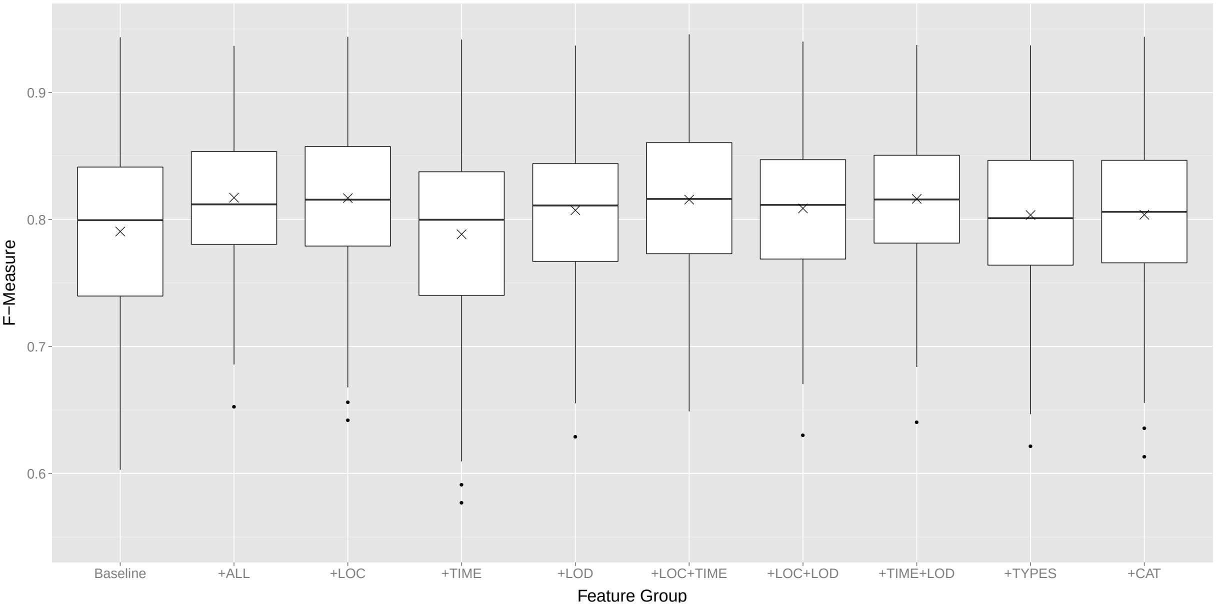


Fig. 8. Box-Whisker diagram for the aggregated performance from holdout sampling with two classes. The mean is indicated by a cross.

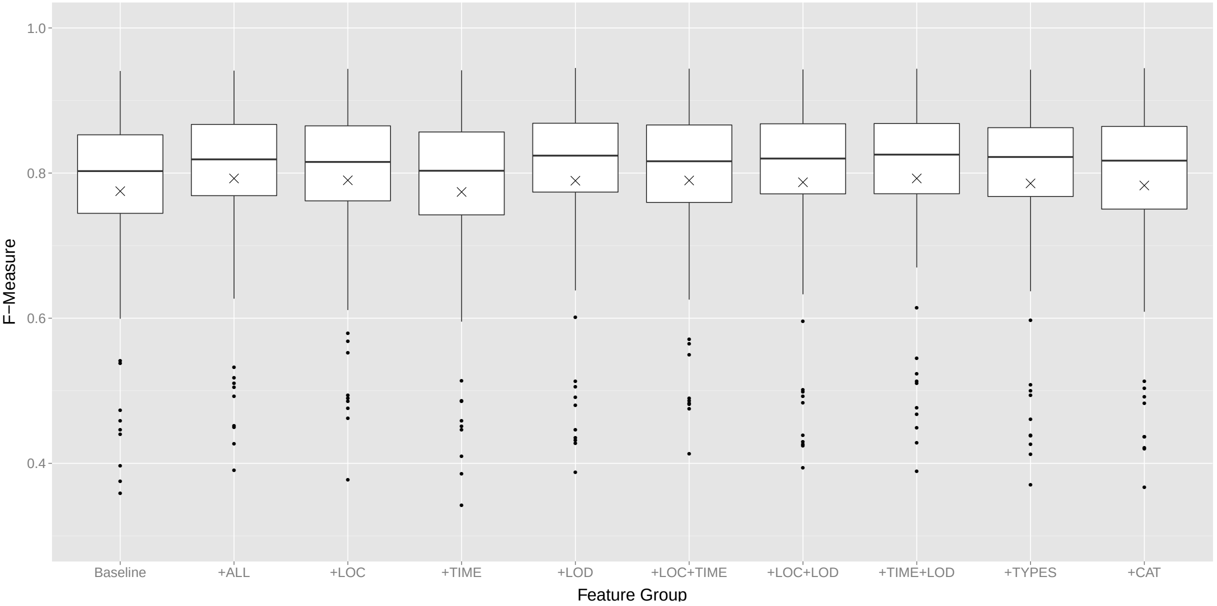


Fig. 9. Box-Whisker diagram for the aggregated performance from the holdout sampling with four classes. The mean is indicated by a cross.

Table 10

P-values from the Nemenyi test for the aggregated performance from holdout sampling with two classes.

	Baseline	+ALL	+LOC	+TIME	+LOD	+LOC+TIME	+LOC+LOD	+TIME+LOD	+TYPES
+ALL	$<$ 0.01***								
+LOC	$<$ 0.01***	0.994							
+TIME	0.998	$<$ 0.01***	$<$ 0.01***						
+LOD	$<$ 0.01***	$<$ 0.01***	0.034**	$<$ 0.01***					
+LOC+TIME	$<$ 0.01***	0.953	1.000	$<$ 0.01***	0.094*				
+LOC+LOD	$<$ 0.01***	0.052*	0.463	$<$ 0.01***	0.985	0.705			
+TIME+LOD	$<$ 0.01***	1.000	1.000	$<$ 0.01***	0.007***	0.998	0.191		
+TYPES	0.018**	$<$ 0.01***	0.002***	$<$ 0.01***	0.999	0.009***	0.713	$<$ 0.01***	
+CAT	0.019**	$<$ 0.01***	0.002***	$<$ 0.01***	0.999	0.008***	0.705	$<$ 0.01***	1.000

Table 11

P-values from the Nemenyi test for the aggregated performance from holdout sampling with four classes.

	Baseline	+ALL	+LOC	+TIME	+LOD	+LOC+TIME	+LOC+LOD	+TIME+LOD	+TYPES
+ALL	$<$ 0.01***								
+LOC	$<$ 0.01***	0.348							
+TIME	1.000	$<$ 0.01***	$<$ 0.01***						
+LOD	$<$ 0.01***	0.953	0.988	$<$ 0.01***					
+LOC+TIME	$<$ 0.01***	0.225	1.000	$<$ 0.01***	0.958				
+LOC+LOD	$<$ 0.01***	0.085*	1.000	$<$ 0.01***	0.810	1.000			
+TIME+LOD	$<$ 0.01***	1.000	0.380	$<$ 0.01***	0.963	0.249	0.097*		
+TYPES	$<$ 0.01***	0.060*	0.999	$<$ 0.01***	0.737	1.000	1.000	0.069*	
+CAT	0.014**	$<$ 0.01***	0.412	0.001***	0.030**	0.568	0.824	$<$ 0.01***	0.882