# Detection and Analysis of Disaster-Related Tweets

Daniel Solomon
solomond@mail.tau.ac.il

Gal Ron
galr1@mail.tau.ac.il

Omri Ben-Horin
omribenhorin@mail.tau.ac.il

## Abstract

TODO: bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla bla.

## 1 Introduction

The popular microblogging service Twitter is a fruitful source of user-created content. With hundreds of millions of new tweets every day, Twitter has become a probe to human behavior and opinions from around the globe. The Twitter 'corpus' reflects political and social trends, popular culture, global and local happenings, and more. In addition, tweets are easy to access and aggregate in real-time. Therefore, we experience an increased interest in natural language processing research of Twitter data.

As one of the world's most widely used social networks, Twitter is an effective channel of communication and plays an important role during a crisis or emergency. The live stream of tweets can be used to identify reports and calls for help in emergency situations, such as accidents, violent crimes, natural disasters and terror attacks (which we all refer to as 'disasters' in this paper).

In this work we utilize techniques from the natural language processing pipeline (tokenization, part-of-speech tagging and named-entity recognition) to work on Twitter data, as opposed to traditional corpora, in order to detect and analyze disaster-related tweets.

**The Dataset** We present our experiments on a dataset of 10,877 tweets[1], labeled to *'disaster-related'* and *'not disaster-related'* with confidence in the range $[0, 1]$. For example, the following tweet is *'disaster-related'* with confidence 1,

> Thunderstorms with little rain expected in Central California. High fire danger. #weather #cawx http://t.co/A5GNzbuSqq

while the following tweet is *'not disaster-related'* with confidence 0.59,

> It's been raining since you left me // Now I'm drowning in the flood // You see I've always been a fighter // But without you I give up

Even for one who is not familiar with the Bon Jovi lyrics in the latter tweet, it is clear that the tweet does not refer to a real natural disaster. However, this observation is hard to make examining only the vocabulary used; the latter tweet contains a variety of 'disastrous' words (e.g. raining, drowning, flood, fighter). This example hints that in order to reach meaningful results we may have to examine additional linguistic features of colloquial writing, as well as Twitter-specific features such as hashtags (#), user at-mentions (@), internet links and emoticons.

**Our Contribution** In this paper we present our work tackling three missions involving disaster-related tweets.

The first mission is *identification* of disaster-related tweets among a variety of tweets. We implemented several classifiers, the best of which achieved TODO: % accuracy on the dataset. We note that this method could have easily been adjusted to identify tweets related to themes other than disasters (e.g. politics-related, sports-related, etc.), given the appropriate dataset.

The second mission is binary classification of disaster-related tweets to one of two categories, *subjective tweets* (i.e. tweets that express an emotion) vs. *objective tweets* (such as news reports on disasters). To achieve this we manually tagged 2,410 disaster-related tweets. The motivation behind this task is that objective tweets like informative news reports are likely to be published after the event had already become clear to emergency services, while subjective tweets may contain invaluable first-person testimonies of ongoing events.

Finally, we extracted named entities to enrich our knowledge on the disaster (mostly location)... TODO: Omri - short description of method and achievements.

To demonstrate our framework we aggregated recent tweets from various locations in the US, extracted disaster-related tweets using our classifier, and then used named-entity recognition to discover entities related to ongoing disasters. For example, ”Hurricane Harvey” appeared as a top named-entity among recent tweets sent from Houston, TX, which we identified as *disaster-related*.

The code of our project is available at `https://github.com/glrn/nlp-disaster-analysis`.

### 1.1 Twitter vs. Traditional Corpora

Tweets are limited to 140 characters and are widely used by non-professional writers. Therefore, Tweet datasets have some unique features that differ from traditional corpora (such as WSJ corpus). These features should be addressed when implementing natural language processing techniques.

First, the grammar of tweets is quite different from edited news text. It is common that tweets are written as colloquial sentences in first person where the subject (’I’) is omitted, as in: ’see the flames from my window OMG’.

Tweets are also characterized by extensive use of abbreviations and slang specific to social-media (e.g. ily for ’I love you’, iono for ’I don’t know). Such abbreviations may squash several parts-of-speech into one token, which poses a challenge to POS tagging.

In addition, due to the colloquial nature of user-created content, it is common that proper words are replaced by phonetically or morphologically similar ones (e.g. ’wtchng’ instead of ’watching’, ’gr8’ instead of ’great’). Users may also use capitalization irregularities, deliberate spelling errors, punctuation irregularities and interjections as a means to express their sentiment, as in the following tweet:

> Haha South Tampa is getting flooded hah- WAIT A SECOND I LIVE IN SOUTH TAMPA WHAT AM I GONNA DO WHAT AM I GONNA DO FVCK #flooding

Lastly, tweets may contain a variety of tokens seen mainly in Twitter and other social media, such as: URLs; emoticons; Twitter hashtags, of the form #tagname, which the authoer may supply to label a tweet; Twitter at-mentions of the form @user which link to other Twitter users; and Twitter discourse functions such as RT (”re-tweet”), indicating that a tweet was originally posted by some other Twitter user, or ellipsis dots (...) at the end (or beginning) of a tweet, indicating that the tweet will be continued in a subsequent tweet by the same user. We note that hashtags and at-mentions can also serve as words or phrases within a tweet, as in:

> Heard about the #earthquake, stay safe everyone.

Regarding URLs, all links posted in tweets are shortened using Twitter’s link service, `http://t.co`, and are converted to a seemingly random 23 characters URL that redirects to the original web address.

## 2 Analysis Wokrflow

**keywords** TODO

- A
- B
- C

TODO: (Gal) Complete this section

## 3 Classification of Disaster-Related Tweets

In the first part of our work we developed a classifier that identifies *disaster-related* tweets from *not disaster-related* tweets, trained on a dataset of 10,877 labeled tweets (TODO: note: maybe better to say that the dataset was 2400 tweets, since we looked only at tweets with confident $> 0.9$) [TODO: Gal: Notice that using 0.9+ confidence filtered around 5000 tweets I think, 2400 were positive...]. We experimented with a Naive Bayes (NB), random forest (RF) and support vector machine (SVM) classifiers.

**Naive Bayes** A supervised probabilistic learning method that classify according to *maximum a posteriori*. In this method we used *unigram* and *bigram* features.

**Random Forest** An ensemble learning method that uses some estimators (trees) to classify. In this method we used *unigram* and *bigram* features.

**Support Vector Machine (SVM)** A discriminative learning method that attempts to find the hyperplane that gives maximum margin between positive and negative classification on the training data, using penalty method for mistakes. In this method we used *unigram*, *bigram*, *tweets metadata* and *POS tagging* features.

## 3.1 Feature Extraction

To train the classifiers we extracted several features for each tweet:

- **Unigrams and bigrams** of tokens in tweet; we used a Python version of `Twokenizer`, tokenizer for Twitter data (Gimpel et al. [2], Myle Ott, 2013 [3]).
- **Tweet metadata**; hashtags, at-mentions and URLs; we parsed the tweet, crawled URLs and used the referred webpage title as supplementary information.
- **Part-of-speech (POS) tags** (bigram and unigram); we used a twitter-specific tagset and tagger presented by Gimpel et al. [2].

**Tokenization** Splitting a tweet to tokens (separated ordered words) is hard due to irregular punctuation patterns and use of punctuation marks for emoticons. For example, the correct tokenization of 'hello (#hashtag)' is to the four tokens [ hello , ( , #hashtag , ) ], but 'hello (:' should be split only to the tuple [ hello , (: ].

`Twokenizer` is a tokenizer designed for Twitter text in English that addresses these issues. It was originally developed in Python by O'Connor et al., 2010 [1], then improved and ported to Java by Gimpel et al., 2011 [2] and later ported back to Python [3]. We use the last version by Myle Ott.

**Metadata Extraction** The hashtags, at-mentions, URLs and emoticons in a tweet carry information that may help better understand the subject and context. Therefore, for each tweet we created a vector of the following metadata features, which we found to be the most expressive:

- Does the tweet contain any links, and if so how many?
- Is '`https://twitter.com`' one of the links (i.e. a link that refers to another tweet)?
- Does the tweet contain a user at-mention (@)?
- Does the tweet contain any hashtags (#), and if so how many?
- Does the tweet contain a happy emoticon (e.g. :D)?

In addition to these features, we extracted information from hashtags and URLs. We attempted to split hashtgs to separate words looking for a CamelCase pattern (for example, #JeSuisCharlie → 'Je Suis Charlie') or words separated by underline. We note that this method is not exhaustive since Twitter users tend to create hashtags composed of joined words, all lower-case.

We also found that the domain name of the URLs in a tweet may give an indication to whether the tweet is disaster-related or not (for example, a link to `https://9gag.com` is a negative hint). However, Twitter applies URL shortening on links, so for each shortened link in the dataset we attempted to reach the original Internet address. We also collected the HTML page title, which often contains the title of an article (for example, in news sites). We managed to expand 4,823 URLs out of 6,157 in the dataset.

For each tweet in the dataset we created an *'extended' version* where (1) hashtags are extracted, (2) URLs are expanded to original URL followed by the page title and (3) every user at-mention is replaced by the token __USERREF__. For example, the following is a tweet,

> http://t.co/c1H7JECFrV @RoyalCarribean do your passengers know about the mass murder that takes place in the #FaroeIslands every year?

and its *extended version* is,

> https://www.royalcaribbean.co.uk/ Holiday Destinations - Cruise Destinations — Royal Caribbean UK. __USERREF__ do your passengers know about the mass murder that takes place in the Faroe Islands every year?

**Twitter POS Tagging** TODO: Gal [2]

## 3.2 Results

Dataset was divided into train (90%) and test (10%) sets. Each classification method was tested with *unigram* features and *bigram* features separately, for the *SVM* classifier, *tweets metadata* and *POS tagging* features were tested as well.
Accuracy was calculated by 3 measures:

**Total Accuracy** number of right classifications out of the total queries ($\frac{TP+TN}{TP+FP+TN+FN}$).

**Positive Predictive Value** number of right positive classifications out of total positive classifications ($\frac{TP}{TP+FP}$).

**Negative Predictive Value** number of right negative classifications out of total negative classifications ($\frac{TN}{TN+FN}$).

For both *Random Forest* and *SVM* classifiers we have tuned parameters using *grid search*. For *Random Forest* we have tuned the number of estimators and for *SVM* we have tuned the penalty constant.

**Random Forest** For the *Random Forest* results 1, we can observe easily that the *Unigram* features have better accuracy and better *NPV*, while *Bigram* features have better *PPV* for any number of estimators. In addition, best result for total accuracy, *PPV*, *NPV* are achieved with 128, 64 and 128 estimators respectively.

**SVM** For the *SVM* results 2 3, we see that for *Unigram* features with or without *POS*, all measurements are becoming constant for $C \geq 10^4$ (penalty is too big to be used). In addition, best result for total accuracy, *PPV*, *NPV* are achieved penalty constant of value $10^4$, $10^3$ and $10^4$ respectively. For *Bigram* features results are very familiar except for a minor gap (around 1.5%) between using *POS* and not. Best result for total accuracy, *PPV*, *NPV* are exactly the same as in the *Unigram* features. Notice that all classifiers used *tweets metadata* features as well.

The following tables 1 2 presents the best measurement result for each classifier, best total accuracy result (94.5%) is achieved with *SVM* classifier using *Bigram*, *tweets metada* and *POS-tagging* features.
As it can be seen, we have succeeded in classifying disaster related or not tweets with several very accurate techniques.
It is important to mention that although the accuracies are very high using simple methods, before preprocessing the tweets, partial measurements were taken and all accuracies were around 82%, therefore preprocessing the tweets was a major improvement for the classification (over 10%).

Table 1: Disaster Classification Best Results: Naive Bayes and Random Forest

| | **Uni NB** | **Bi NB** | **Uni RF** | **Bi RF** |
|---|---|---|---|---|
| accuracy | 0.921 | 0.864 | 0.911 | 0.890 |
| ppv | 0.977 | 1.000 | 0.945 | 0.969 |
| npv | 0.891 | 0.813 | 0.892 | 0.856 |

Table 2: Disaster Classification Best Results: SVM

| | **Uni SVM** | **Uni POS SVM** | **Bi SVM** | **Bi POS SVM** |
|---|---|---|---|---|
| accuracy | 0.935 | 0.937 | 0.933 | 0.945 |
| ppv | 0.961 | 0.946 | 0.952 | 0.955 |
| npv | 0.935 | 0.938 | 0.926 | 0.939 |

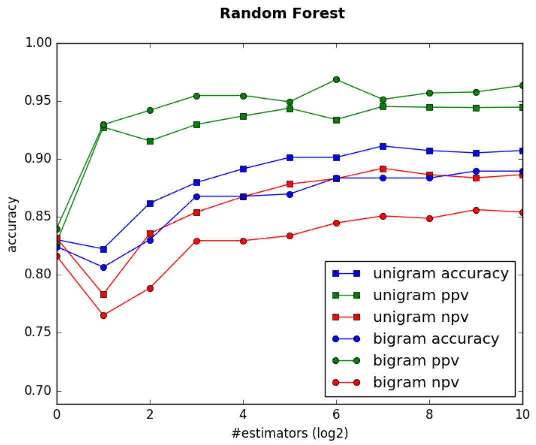

Figure 1: Disaster Classification Random Forest (Unigram vs. Bigram)

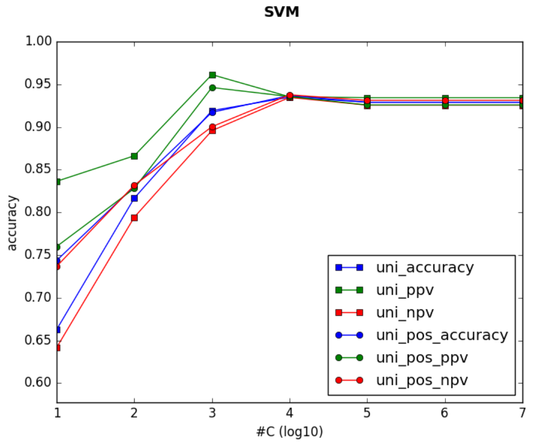

Figure 2: Disaster Classification SVM (Unigram vs. Unigram and POS)

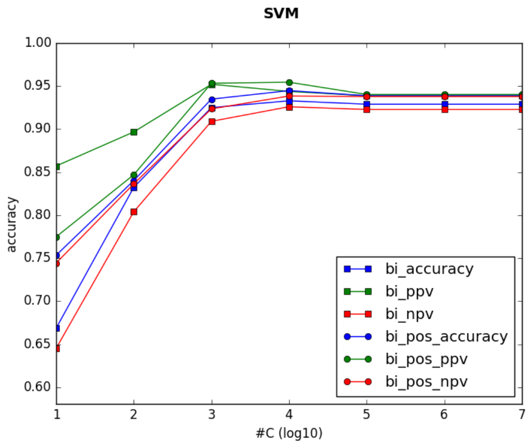

Figure 3: Disaster Classification SVM (Bigram vs. Bigram and POS)

## 4 Sentiment Analysis of Tweets

Sentiment analysis is considered to be a very interesting task in *NLP*. It aims to determine the writer's attitude. Sentiment analysis tasks are usually target for multiclass classification such as feelings extraction (*”sadness”*, *”happiness”*, *”anger”*, etc.). Another target may be classifying the *polarity* of a given text (for example *”positive”* or *”negative”*).
We concentrated on the polarity task of classifying *’disaster-related’* tweets as *”objective”* or *”subjective”*. In general vision, the ability of classifying *’disaster-related’* tweets into these two categories may be used to extract facts and raise the confidence level of the disaster. Although sentiment analysis is considered as a interesting task, it is surely a difficult one, combining it with tweets as dataset results an even more complicated task, since for the attributes we have spoken about in section 1.1. For example the following tweet is *’disaster-related’*, but it is hard to tell whether it is objective or not:

> Thunder lightening torrential rain and a power cut!

### 4.1 The Dataset

Since supervised classification is requested, we needed a dataset containing labels of this polarity. However, we could not find any dataset *’disaster related’* or not tweets with a *objective/subjective* label. Therefore we extracted only *’disaster related’* tweets from our original dataset (with confidence level $\geq 0.9$) and labeled them manually according to the following rule:

- **Objective** is an undeniable truth. It is universally agreed upon. No logical person can deny that.
- **Subjective** is an opinion. The beauty is in the eye of the beholder.

We labeled 2,100 *’disaster related’* tweets as *objective/subjective* manually. Each tweet was labeled by two persons, on agreement, the label was kept, on disagreement, a third person broke the tie. Around 80% of the tweets are objective while only 20% are subjective.

### 4.2 Feature Extraction

We assumed that this task can achieve high accuracy without even using the words themselves (n-grams and meanings), but using the tweet structure, peripheral data and *POS*-tagging.
To train the classifiers we extracted the following features for each tweet:

- Number of exclamation marks.
- Presence of exclamation mark.
- Number of question marks.
- Presence of question mark.
- Presence of *URL*.
- Presence of emoticon (using *emoticon.py* extractor from *Twitter NLP* framework [4]).
- Number of digits.
- Number of capital words.
- Number of capital letters.
- Number of punctuation marks and symbols (!”#$%&’()*+,-./:;¡=¿?@[\]ˆ_‘{}˜).
- Tweet’s length.
- *POS*-tagging:
  - Number of adjectives.
  - Number of verbs.
  - Number of adverbs.
  - Number of pronouns.
  - Number of proper nouns.
  - Number of possessive endings.

### 4.3 Results

We trained *SVM* and *Random Forest* classifiers for this job. Using $C = 10^4$ as a penalty constant for *SVM* and 128 estimators for *Random Forest*.
Once again dataset was divide into train (90%) and test (10%) sets. Each classification method was tested with the previous mentioned features, using feature selection with *ANOVA F-test* model. We repeated the test each time with more top $K$ features selected by the feature selection model (out of 18 features total) and measured accuracies again. The goal was to find which features are the most important for our task. Measurements are total accuracy, *PPV* and *NPV*.

**Random Forest** For the *Random Forest* results 4, we can see that best total accuracy is achieved with 13 features, so for *NPV*, but best *PPV* is achieved with only 1 feature (might be because of the biased dataset). The most influence features which were used are: number of exclamation marks, presence of exclamation mark , presence of question mark, presence of URL, presence of emoticon, number of digits, number of capital letters, number of punctuation marks and symbols, tweeter's length.

**SVM** For the *Random Forest* results 5, we can see that best total accuracy is achieved with 4 features only, so for *NPV*, but best *PPV* is achieved with only 1 feature as well (again, might be because of the biased dataset). The most influence features which were used are: number of exclamation marks, presence of URL, presence of emoticon, number of punctuation marks and symbols.

The following tables 3 presents the best measurement result for each classifier (number of features used can be seen in column title), best total accuracy result (87.1%) is achieved with *Random Forest* classifier using 13 features.

Table 3: Objective/Subjective Classification Best Results

| | RF (1) | RF (13) | SVM (1) | SVM (4) |
|---|---|---|---|---|
| accuracy | 0.805 | 0.871 | 0.805 | 0.848 |
| ppv | 0.905 | 0.890 | 0.905 | 0.874 |
| npv | 0.500 | 0.750 | 0.500 | 0.667 |

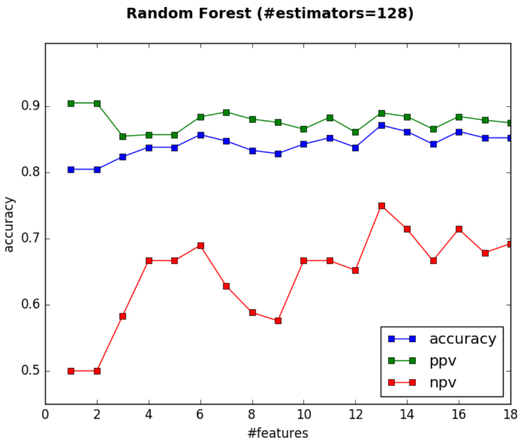

Figure 4: Objective/Subjective Classification Random Forest

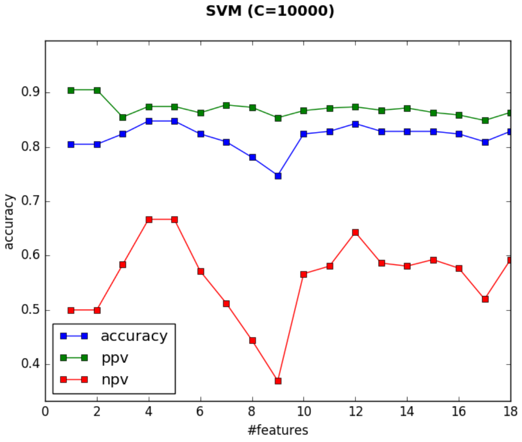

Figure 5: Objective/Subjective SVM

## 5 Named-Entity Recognition in Tweets

In order to extract relevant information from disaster tweets, we have decided to use named entity recognition methods. NER allows us to classify people, organizations, locations and time expressions, which we can combine with our previous results for exposing relevant information on disasters.

**GMB** [2] In order to use NER, we used the Groningen Meaning Bank - a large corpus, composed of thousands of texts in raw and tokenized format, tags for part of speech, named entities and lexical categories, and discourse representation structures compatible with first-order logic. The top level categories of the corpus are:

- geo - Geographical Entity
- org - Organization

- per - Person
- gpe - Geopolitical Entity
- tim - Time indicator
- art - Artifact
- eve - Event
- nat - Natural Phenomenon

We used only the top categories, since they are sufficient for extracting relevant information on disasters from the tweets. The tags of the GMB corpus are translated to a NLTK compatible format, for using it combined with our previous results.

For tagging the named entities, a classifier based tagger can be used. We chose, for this part, the standard NLTK classifier. The best features of the tagger are unigrams, part of speech tagging, stemmer lemma of the current word, same attributes of the previous and next words, the previous tagging, and whether the current word is capitalized or contains special characters.

Tweets are often disorganized and does not contain structured sentences. In order to reduce errors, each tweet is treated as non-capitalized, and is stripped of URLs and special characters.

**Results** Three main categories of the GMB corpus remained relevant in the Tweeter world - location, organization, and geopolitical entity. A fourth category, time, was somewhat ambiguous - as it described both time entities and irrelevant ones. Other categories did not manage to reach any level of internal agreement on any entity.

By using the GMB corpus as the training data, we performed named entity recognition on the previously used dataset, of 10,877 labeled tweets. From those tweets, we singled out only those who were labeled as disasters. The main results were:

- Geographical Entity - California and Hiroshima were the dominant entities, as there were tweets about disasters in those places, followed by Japan, India, south, and Saudi Arabia.
- Organization - This category was composed by news companies, such as ABC, and by examples such as Obama, which was involved in the California disaster. Many user references are marked as an organization as well.
- Geopolitical Entity - Composed mostly of user references, and nationalities - Israeli, Turkish and much more.
- Time Indicator - ranged from a description of the current time - wildfire or typhoon, to description of the 70th anniversary of Hiroshima atom bombing.

From those results, we can learn the location, nationality of involvement, associated organizations, and the type of the disaster.

## 6 Experimenting with Recent Tweets

TODO: (Gal) Complete this section

**keywords** Twitter's Search API

## 7 Conclusions

**Future work** TODO

## References

[1] Brendan O'Connor, M. K., and Ahn, D. Tweetmotif: Exploratory search and topic summarization for twitter. In *Proc. ICWSM* (2010).

[2] Kevin Gimpel, Nathan Schneider, B. O. D. D. D. M. J. E. M. H. D. Y. J. F., and Smith, N. A. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proc. ACL* (2011).

[3] Ott, M. Python port of `Twokenizer`. `https://github.com/myleott/ark-twokenize-py`, 2013.

[4] Ritter, A. Twitter nlp tools. `https://github.com/aritter/twitter_nlp`, 2011.

## Notes

[1] **"Disasters on social media" Dataset by CrowdFlower**: Contributors looked at over 10,000 tweets culled with a variety of searches like "ablaze", "quarantine", and "pandemonium", then noted whether the tweet referred to a disaster. `https://www.crowdflower.com/wp-content/uploads/2016/03/socialmedia-disaster-tweets-DFE.csv`

[2] **"Groningen Meaning Bank" Dataset by University of Groningen**: comprises thousands of texts in raw and tokenised format, tags for part of speech, named entities and lexical categories, and discourse representation structures compatible with first-order logic. `http://gmb.let.rug.nl/`