Contents

1 Introduction (5p)

1.1 Twitter

Twitter ist ein seit 2006 bestehender und heute weltweit genutzter Webdienst zur Versendung von Kurznachrichten (sogenanntes Mikroblogging). Nur ein kleiner Teil der Nachrichten, der sogenannten Tweets, ist in deutscher Sprache verfasst. Die überwiegend genutzten Sprachen sind Englisch, Spanisch, (...). Auch im relativen Vergleich zur Einwohnerzahl zeigt sich, dass in Deutschland, Österreich und der Schweiz wenig getwittert wird, während das Netzwerk etwa in den Niederlanden, Großbritannien, Japan und Indonesien extrem populär ist.

Auch innerhalb Deutschlands ist die räumliche Verteilung versendeter Tweets auf Grundlage mitgesendeter Geodaten ermittelbar. Sie spiegelt größtenteils die Bevölkerungsverteilung wider; Zentren sind vor allem Berlin und das Ruhrgebiet. Allerdings ist – neben dem Twittern allgemein – im Besonderen das Mitsenden von Geodaten (sogenanntes Geotagging) im deutschsprachigen Raum eher unpopulär.

Es gibt drei grundsätzlich verschiedene ortsbezogene Informationen, die bei einem Tweet mitgesendet werden können. Keine der Angaben ist für den Nutzer verpflichtend, sodass nicht zu allen Tweets ortsbezogene Daten verfügbar sind. Zunächst kann der Nutzer in seinem Profil einen Standort eingeben. Er erscheint im JSON-Objekt des Tweets als Feld *location* im Unterobjekt *user*. In unserem Korpus ist zu x% der Tweets diese Angabe vorhanden. Allerdings werden hier auch sehr gerne Fantasie-Orte eingetragen. Hinzu kommt die mögliche Mehrdeutigkeit und Ungenauigkeit der Angaben, weswegen sie für maschinelle Verwertung praktisch ausscheiden. Beim Absenden eines Tweets besteht außerdem die Option *Standort hinzufügen*. Dort kann ein eindeutiger, benannter Ort hinzugefügt werden. Im JSON-Objekt erscheint er als Unterobjekt *place*, unter anderem mit Namen, Typ (bspw. "city" für Stadt), Land und einer *bounding box*, einem Rechteck aus vier Geokoordinaten, welches den Ort einschließt. "Tweets associated with places are not necessarily issued from that location but could also potentially be *about* that location." Diese Angabe findet sich in unserem Korpus zu x% der Tweets. Schließlich ist es möglich, von GPS-fähigen Geräten aus direkt den tatsächlichen Absendeort des Tweets mitzusenden. Die Koordinaten werden als Unterobjekt *geo* im JSON gespeichert. x% der Tweets in unserem Korpus tragen diese Angaben.

1.2 General Idea

Ziel: ungefähre regionale Einordnung eines Tweets innerhalb des deutschssprachigen Raums trotz der o. g. seltenen Geodaten und schlecht benutzbaren Herkunftsangaben
Idee: Sprache verrät Herkunft, also sollte aus dem reinen Tweettext die Herkunft ablesbar sein

Es existieren hier zwei unterschiedliche Größen. Zum einen gibt es den oder die Orte, an denen ein Twitterer aufgewachsen ist und die ihn sprachlich geprägt haben. Hauptsächlich diese Orte messen wir, wenn wir nach mundartlichen Ausdrücken und Ausdrücken der regionalen Alltagssprache suchen. Auf der anderen Seite steht der momentane, mitunter sehr kurzfristige Aufenthaltsort, von dem aus der Nutzer twittert. Auf Inhaltsebene

$$sim(\vec{q}, \vec{d_j}) = \frac{\sum_{i=1}^{N} w_{i,q} \times w_{i,j}}{\sqrt{\sum_{i=1}^{N} w_{i,q}^2} \times \sqrt{\sum_{i=1}^{N} w_{i,j}^2}}$$

Figure 1: Cosine similarity

der Tweets wird er sich eher in ortsbezogenen Begriffen (Ortsnamen, Verkehrsknotenpunkte, Lokalitäten, lokale Ereignisse und Persönlichkeiten etc.) widerspiegeln. Er ist es außerdem, den wir aus den Geodaten von Tweets erfahren. Nun sind Geodaten jedoch die einzigen Daten, die wir zur Evaluierung unserer Ergebnisse verwenden können. Während unser geodatengestützter Ansatz damit recht passend evaluiert werden kann, zielt unser Regiowort-Ansatz speziell auf die Größe *sprachliche Herkunft des Twitterers* ab und wird die Evaluierung daher zwangsläufig mit einem gewissen Handicap absolvieren.

Ansatz:

- Einteilung des deutschsprachigen Raums in Regionen
- Machine Learning auf Trainingsdaten aus diesen Regionen
- Bag-of-words model (Betrachtung von Unigrammen)

1.3 Regional word attempt

1.4 Geo location attempt

1.5 Expectations

1.6 Sources, used corpora

1.7 Regions

Die Grundüberlegungen, auf denen unsere konkrete Einteilung des deutschen Sprachraumes in Regionen basiert, gab hauptsächlich der Regiowort-Ansatz vor. Die Datenlage im Atlas der deutschen Alltagssprache zeigte uns, in welcher Größenordnung man den Sprachraum auf Basis regionaler Alltagssprache einteilen kann. Auch die konkrete Festlegung der einzelnen Regionen trafen wir beim Sichten der dortigen Daten. Dennoch trug hier auch Twitter als unser Anwendungsbereich ein entscheidendes Kriterium bei: Es sollten Regionen entstehen, in denen jeweils mit einem genügend großen Aufkommen von Tweets zu rechnen war. So zeigte sich in den Daten des Atlas der deutschen Alltagssprache auch eine kleinere Region im Gebiet Saarland/Luxemburg mit charakteristischen Eigenheiten, die wir allerdings wegen zu geringer erwartbarer Menge von Tweets nicht übernahmen.

$$q = \text{'Ich glaube @Drahflow tippt noch schneller als er redet. ;) \#om13'}$$
$$\vec{q} = (0.427, 0.38, 0.4, 0.39, 0.391, 0.602, 0.41)$$
$$\vec{\bar{d}} = \frac{\sum_{i=0}^{N} d_i}{N} = (0.376, 0.336, 0.349, 0.346, 0.361, 0.481, 0.379)$$
$$sim(\vec{q}, \vec{\bar{d}}) = 0.9986$$

Figure 2: Example for the similarity calculation

2 Algorithms

2.1 Cosine similarity

Although Tweets may differ from one region to another in some way, a huge percentage of the German Twitter users write their messages exclusively in standard German with no signs of any regional influence whatsoever.
For example the following Tweet just appeared in my timeline:

'Ich glaube @Drahflow tippt noch schneller als er redet. ;) #om13'

Keeping in mind that usernames, hashtags, smileys and punctation are being removed in the pre-processing, this Tweet contains way to ordinary words to be assigned to a specific region. It is written in pure High German and therefore could be sent from a town in Bavaria as well as from Berlin and depending on the data we use to train our algorithm with, the result of the program could be 'Austria' or 'Northern Germany' as well.

To face this problem we decided to filter this kind of undistinguished Tweets to have more reliable results and because of that also increase the accuracy of the algorithm.
Our idea was to determine the average Tweet-vector $\vec{\bar{d}}$ of all the Tweets in the training-corpus and to compare the Tweet-vector $\vec{q}$ of a given Tweet with it, using the cosine metric (see Figure 1 on page 3) as recommended in Jurafsky & Martin. If the similarity between both vectors is smaller than a specified threshold, we continue to map the Tweet to one of the region, if not we stop and return a message, that the Tweet is written in standard German and can not be classified. In the example in figure 2 on page 3, the vector $\vec{q}$ for the Tweet mentioned above is compared to the average Tweet-vector $\vec{\bar{d}}$, returning a very high similarity of 0.9986. Nevertheless the algorithm states, that the Tweet was most likely sent from Switzerland.

But the most difficult thing was to find the right threshold, that separates the too ordinary Tweets from the regional ones. To make a guess which percentage of all Tweets are written in standard German, we calculated the cosine similarity of all Tweets in the training-corpus to the average vector, sorted them and had a look at the distribution (figure 3 in page 5).
The result was not surprising: More than 80% of all Tweets had a similarity of 0.9 or

Figure 3: Cosine similarity between all Tweet-vectors and the average Tweet-vector

higher. Or, seen from another point of view, only 20% of all Tweets differ enough from the average vector to calculate reliable results.

In the experiments in the sections 3 (regional based attempt) and 4 (geo location based) we tested different thresholds to find a good balance between a reliable classification with a high accuracy and the coverage of as many Tweets as possible.

3 Regional word attempt (8p)

3.1 Detailled description of the idea

3.2 Source of the data and creation of the CSV

3.3 Why the loops are so important

3.4 Experiments

3.4.1 Parameters

3.4.2 Expectations

3.4.3 Discussion

3.4.4 -> new Experiment

3.5 Conclusion

4 Geo location attempt

As stated in the previous chapter, the foundation of all calculations in the regional word attempt is a list of a few hundred words that appear more likely in a specific region. Although this list and their probability distribution is based on scientific research it markes the weak spot of this attempt for number of reasons. For example people in a specific region use a typically word in their everyday language while speaking to their friends or family, but there is no proof that this people also use this words in their written language, even if it is only their private Twitter account. In addition, the list is way to short to cover only fraction of the words, people use on Twitter, so the end results mostly rely on the data that is generated in main loop of the algorithm.

We seem to have no other choice but to trust this generated values, so we had the idea of skipping the manually created word list and use an automatically created based on a corpus of Tweets with a geo location instead.
Before entering the main loop, we had to write another algorithm that learns the distribution on all seven regions for all the words in the corpus. This way we are generating

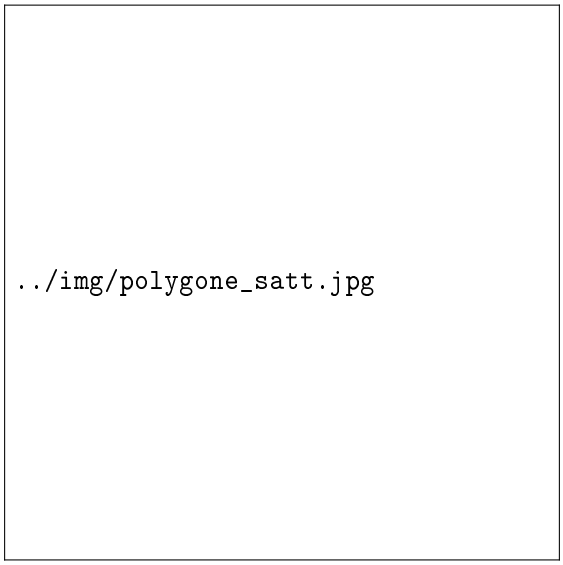

Figure 4: Map of the regions and their index represented as polygons

a list of words that covers nearly all the words. In order to classify a tweet, that has a geo location, we had struggled to find a good way to represent the seven regions in a way, that we could easily check from which of them a Tweet was sent. After a few unsuccessful approaches, we decided to define polygons for the regions that are not overlapping each other, but also leave no gaps between them. For the value of the points we simply used their longitude and latitude coordinates, that can be represented as floats. We did not implement a point-in-polygon algorithm ourself, but used the version found here [SOURCE] instead. To find out from which region a tweet was sent, we iterate over all regions and return the first one, where the point-in-polygon function returns true. In this first step, the Tweet-vector consists only of zeros and a one for the feature standing for the region it was sent from.

Let's assume that the coordinates of some Tweet reveal, that its region is "Westdeutschland", represented by the index 3. Therefore the Tweet-vector is $(0, 0, 1, 0, 0, 0, 0)$. The next step consists of iterating over all token in the tweet and add the Tweet-vector to their word-vectors. To put it simply, we increase the counter for the specific feature of all occurring tokens by one. In the final step, we iterate over all word-vectors and use our `normalize()`-function to attain the probability distribution, how likely it is that the token is used in a specific region. Because of this normalization, the format of the outcome is comparable to the input list in the regional-word attempt and we can use the main part of the algorithm without having to adjust it.

Example Tweet: $d =$ 'Hello Twitter!' from the region 'Westdeutschland'.

$$\begin{aligned}
\vec{t}_{hello} &= (1,3,2,5,2,1,0) \\
\vec{t}_{twitter} &= (0,0,0,0,0,0,0) \\
\vec{d} &= (0,0,1,0,0,0,0) \\
\vec{t}'_{hello} &= \vec{t}_{hello} + \vec{d} = (1,3,3,5,2,1,0) \\
\vec{t}'_{twitter} &= \vec{t}_{twitter} + \vec{d} = (0,0,1,0,0,0,0) \\
\texttt{normalize}(\vec{t}'_{hello}) &= (0.04, 0.13, 0.13, 0.22, 0009, 0.04, 0) \\
\texttt{normalize}(\vec{t}'_{twitter}) &= (0,0,1,0,0,0,0)
\end{aligned}$$

Figure 5: Example for the creation of the initial word-vectors

4.1 Source of data

The datasets used for the generation of the first word-vectors have been extracted from the Scheffler-corpus using the same classification function as in the main program. Although the corpus mostly consist of German Tweets (filtered with *LangID*), about 20% of the Tweets with geo-location have not been sent from one of the defined regions and therefore have been ignored.
We created balanced sets, where the amount of Tweets are the same for all regions. Because the fewest number of Tweets (8787) came from region 6, 'Austria', we ended up with only 61509 Tweets for all regions.

In order to create a gold-standard, we substracted 150 Tweets from each region, leaving us 60459 for the training.
Our assumption was, that datasets of different sizes lead to different results and we wondered, how it would effect the accuracy, if we would use the same data for the creation of the word-vectors and for the main-algorithm. Therefore we created three balanced sets, one unbalanced set of all geo annotated Tweets and as a reference one huge set of normal Tweets. Of course none of these sets contain any Tweets from the gold-standard-set.

1. *balanced-21k* with 3000 Tweets from each region.
2. *balanced-39k* consisting of the remaining 39456 Tweets
3. *balanced-61k* combining the both previous sets.
4. *geo-175k* with all Tweets from the corpus that were sent from one of the defined regions. Not balanced.
5. *all-1500k* contains 1.5 million mostly not geo annotated Tweets.

4.2 Experiments

4.2.1 Dataset combinations

In order to give a solid foundation which data we should use in further experiments, we started to compare the combination of the five datasets we created. We wondered, how it would affect the accuracy, if we would for example train the geo-algorithm with a small set of geo-annotated data but use a very big set of Tweets for the learning of the word-vectors in the loop of the main-algorithm. Another question was, what results we would get, if the data in the main-algorithm contains Tweets, that have been used in the geo-algorithm before.

Our expectation was in general, that the accuracy would rise, the more data we use and that the reuse of Tweets in the main-algorithm would lower it, because this data already had an effect and using it again would not change anything.

Unsurprisingly, using the smallest set, *balanced-21k*, for the training of the geo-algorithm nearly always produced the lowest accuracy.
But our hypothesis about the reuse of data was falsified: The combination of the *balanced-39k* set for the learning in the geo-algorithm and *balanced-61k* for the main-algorithm generated the third best result with an accuracy of 0.345.

Also we discovered, that the use of balanced data in the geo-algorithm, where all regions are represented by the same amount of documents, is crucial for good results. Having a look at the row for the *unbalanced-175k*, it produces a very bad accuracy. Even in combination with the set containing 1.5 million Tweets (*all-1500k*), it marks the second last rank.

This experiment showed us, that we should use the *balanced-39k*-set for the training of the geo-algorithm and the *geo-175k*-set for the main-algorithm, because it generated the best accuracy of 0.365. We were surprised, that the combination *balanced-39k* / *all-1500k* had a worse result, but interpreted it as a result of the unfiltered data of the *all-1500k*-set.

Geo dataset	balanced-21k	balanced-39k	balanced-61k	geo-175k	all-1500k
balanced-21k	0.319	0.308	0.304	0.307	0.308
balanced-39k	0.306	0.353	**0.345**	**0.368**	0.306
balanced-61k	**0.336**	**0.361**	0.344	0.360	0.332
unbalanced-175k	0.303	0.343	0.322	0.352	**0.340**

Calculation method: *Normalized*; Stopwords: *Top 200*; Loops: *1*; Estimated amount of non-regional Tweets: *60%*, leading to a similarity threshold between *0.999* and *0.991*, depending on the dataset.

Table 1: Comparrision of the combination of all datasets

4.2.2 Guessing the amount of regional Tweets

5 Conclusion (3p)

5.1 Results of both attempts compared

5.2 Reasons for good/bad results

5.3 Outlook: What could be better if we had enought time