DCIT: Discourse Connectives in Twitter

Jessica Grasso jgrasso@uni-potsdam.de
C. Clayton Violand cvioland@uni-potsdam.de

21 August 2015

Abstract

DCIT is a tool written in Python that analyzes the usage of discourse connectives in German Twitter data. It utilizes Twitter text data, parallel POS-tagged text, and the DiMLex discourse connective framework. It allows for the analysis of the use of discourse connectives among Twitter speech, and provides the groundwork for further analysis of this type of dialogue as compared to more standard forms of conversation.

1 Background and Related Work

Discourse markers or discourse connectives are words that show the presence of a discourse relation in text. These words form a closed class, must meet certain fixed criteria, and share some characteristics of content words. These relations are vital to the coherence of a text, and can be used to analyze content and structure [4].

Due to the nature of the Twitter platform, one might expect that the use of these discourse connectives differs when tweeting. In order to test this idea, discourse connective data must first be collected from the tweets. To accomplish this, classic data-comparison techniques are brought in, such as POS-tagging.

Part-of-speech tagging for Twitter data faces many challenges due to the non-standard language common on that platform, including acronyms, missing or shortened words, emoticons, and URLs, in addition to more common out-of-vocabulary items such as proper names. [5]. We cannot comment on which, if any, techniques were used to overcome these challenges, as the output files were provided to us, but this is a limitation to keep in mind.

This project uses DiMLex, a lexicon of German discourse markers developed by the Discourse Lab at the University of Potsdam [4]. From this lexicon we extracted our list of German discourse connectives and additional information about each.

We also used results found in the bachelor thesis of Angela Schneider (University of Potsdam), titled *Disambiguierung von Diskurskonnektoren im Deutschen* [1], as well as a paper published by Schneider and Manfred Stede containing a summery of the same work [2]. The results of these works were used as a first step in disambiguating the potentially ambiguous discourse connectives in this project.

2 System Description

DCIT is a tool written in Python that analyzes the usage of discourse connectives in German Twitter data. Given a list of German discourse markers and one or more files containing German-language tweets, the tool counts possible discourse connectives, performs disambiguation on the ambiguous connectives, and re-counts, printing a summary of the information collected and outputting annotated versions of the tweets.

2.1 Input

- `dimlex.xml` - a lexicon of German discourse markers [4]. At the time of submission the current version was that of 25 July 2015.
- One or more `.xml` files containing tweet threads. This format was provided to us and was not altered for this project. Although not perfect, this data was already cleaned and assumed to contain only German-language tweets from a one-month time period.
- For each file above, a corresponding `.txt` file containing the part-of-speech tagged text of each tweet and the unique ID number of each tweet for identification purposes. For this project, these files were provided to us by Wladimir Sidorenko [6], and are tagged using the STTS tag set [3].

2.2 Output

- for each tweet thread file, a modified version of that file with the following attributes added to the `tweet` tag:
  - `has_dc` - with value `True` or `False` corresponding to whether the tweet contains at least one (potential) discourse connective
  - `num_dcs` - the number of (potential) discourse connectives contained in that tweet Additionally,
  - the tag `DC\` is added before each discourse connective in the tweet text itself

3 System Method

First, the DiMLex lexicon is read in and stored as an object (`get_dcons.py`). Each connective has several important features used in this project:

- continuity - whether the discourse connective has one part (*unterdessen*, *und zwar*) or multiple parts (*um ... willen*, *umso mehr ... als*)
- type - for each part, whether that part is single (*um*, *seitdem*) or phrasal (*umso mehr*, *und zwar*)
- ambiguity - true for the entries that have multiple possible interpretations, either as a discourse connective or another function

The lexicon also contains several additional features that we did not utilize.

Then, the tweets are read in (`get_tweets.py`). Since even one day of tweets is far too much to fit into memory (at least on the machines at our disposal), each tweet is sent through the following pipeline individually, and at no point should more than one file (corresponding to one day) be open simultaneously. Each tweet is represented as an object, which at this point contains only features extracted from the file. These include:

- `id` - unique tweet ID number

– `_original` - original, unmodified tweet text

– `words` - tokenized tweet text

– `raw` - tweet text made lowercase and later further modified as needed (e.g. discourse connective deletion occurs to prevent certain special types from being found twice).

The next step is to determine and save more details about each tweet (`get_matches.py`). This includes searching for all possible discourse connectives, and counting and storing some basic information about how many of each has been found. This information, since it is collected over a large number of tweets, is stored in another object (`get_info.py`) which can be referenced and outputted at a later time. In addition to these counts, a list of tuples containing all found (potential) discourse connectives, their ambiguity status, and their location in the string is populated in each tweet object for later use.

Once this basic information has been gathered, the task is to disambiguate the ambiguous discourse connectives, that is, to determine which are truly discourse connectives and which are likely to be performing other functions in the text (`disambiguate.py`). For this task we used Schneider's results, resulting in three categories. Category 0 includes the words from Table 1 of Schneider & Stede (2012) [2], and entirely eliminates words that occur only very infrequently as discourse connectives from the list of possible connectives. Category 1 includes the words from Table 2.2 of Schneider [1] for which the ambiguous status can be resolved using the part of speech of the connective in question. Category 2 includes the words from Table 3.1 of Schneider [1] for which further context–in addition to the part-of-speech–is needed in order to disambiguate.

After this step, the basic analysis mentioned above is repeated (`post_disambiguation_stats.py`) using a much simplified but otherwise similar method as before. Compared with the initial stats, there should now be both fewer potential discourse connectives (as some will have been removed during disambiguation) as well as fewer ambiguous matches (since all removed were ambiguous, but not all ambiguous could be sorted).

Finally, these results are written to file (`write_results.py`). Currently, the initial files are edited, in that the XML attributes described above are added, along with in-text tags marking the discourse connectives. These resulting files can then be used in further analysis.

4 Results

Here is an example of the results for a subset of the data, roughly 5,000 lines of tweets from 3 April 2013.

Pre-disambiguation

```
Pre-disambiguation
-- SUMMARY --
There are 277 possible Discourse Connectives.
--------------------------------------------------------------------
I. all matches.
--------------------------------------------------------------------
Found 4939 potential Discourse Connective matches amongst 3306 Tweets.
Found a potential Discourse Connective in 2436 out of 3306 Tweets.
Potential Discourse Connective Saturation is 0.736842.
--------------------------------------------------------------------
```

```
of type = 'continuous': 4906
of type = 'discontinuous': 33
--------------------------------------------------------------------
II. ambiguous matches.
--------------------------------------------------------------------
Found 4672 ambiguous cases amongst 4939 matches.
--------------------------------------------------------------------
of type = 'continuous': 4641
of type = 'discontinuous': 31
--------------------------------------------------------------------
III. Most common discourse connectives.
--------------------------------------------------------------------
Printing top 10 continuous connectives:
und occurs  798  times, which is  16.2657969833  percent.
auch occurs  446  times, which is  9.09090909091  percent.
aber occurs  363  times, which is  7.39910313901  percent.
ja occurs  278  times, which is  5.66653077864  percent.
wie occurs  221  times, which is  4.50468813698  percent.
nur occurs  216  times, which is  4.40277211578  percent.
fr occurs  215  times, which is  4.38238891154  percent.
dann occurs  207  times, which is  4.21932327762  percent.
da occurs  190  times, which is  3.87280880554  percent.
mal occurs  185  times, which is  3.77089278435  percent.
--------------------------------------------------------------------
Printing top 10 discontinuous connectives:
wenn  ...  auch occurs  16  times, which is  48.4848484848  percent.
so  ...  dass occurs  10  times, which is  30.303030303  percent.
von  ...  wegen occurs  2  times, which is  6.06060606061  percent.
entweder  ...  oder occurs  2  times, which is  6.06060606061  percent.
sowohl  ...  als auch occurs  1  times, which is  3.0303030303  percent.
weder  ...  noch occurs  1  times, which is  3.0303030303  percent.
auf  ...  hin occurs  1  times, which is  3.0303030303  percent.
teils  ...  teils occurs  0  times, which is  0.0  percent.
umso  ...  als occurs  0  times, which is  0.0  percent.
umso weniger  ...  als occurs  0  times, which is  0.0  percent.
--------------------------------------------------------------------
Printing top 10 ambiguous connectives:
und occurs  798  times, which is  17.0804794521  percent.
auch occurs  446  times, which is  9.54623287671  percent.
aber occurs  363  times, which is  7.76969178082  percent.
ja occurs  278  times, which is  5.95034246575  percent.
wie occurs  221  times, which is  4.73030821918  percent.
nur occurs  216  times, which is  4.62328767123  percent.
fr occurs  215  times, which is  4.60188356164  percent.
dann occurs  207  times, which is  4.43065068493  percent.
da occurs  190  times, which is  4.06678082192  percent.
mal occurs  185  times, which is  3.95976027397  percent.
--------------------------------------------------------------------
```

Post-disambiguation

```
Post-disambiguation
-- SUMMARY --
There are 276 possible Discourse Connectives.
--------------------------------------------------------------------
I. all matches.
--------------------------------------------------------------------
Found 4056 potential Discourse Connective matches amongst 3306 Tweets.
Found a potential Discourse Connective in 2254 out of 3306 Tweets.
Potential Discourse Connective Saturation is 0.681791.
--------------------------------------------------------------------
of type = 'continuous': 4033
of type = 'discontinuous': 23
--------------------------------------------------------------------
II. ambiguous matches.
--------------------------------------------------------------------
Found 3789 ambiguous cases amongst 4056 matches.
--------------------------------------------------------------------
of type = 'continuous': 3768
of type = 'discontinuous': 21
--------------------------------------------------------------------
III. Most common discourse connectives.
--------------------------------------------------------------------
Printing top 10 continuous connectives:
und occurs  798  times, which is  19.7867592363  percent.
aber occurs  363  times, which is  9.00074386313  percent.
ja occurs  278  times, which is  6.89313166377  percent.
wie occurs  221  times, which is  5.47979171832  percent.
fr occurs  215  times, which is  5.33101909249  percent.
dann occurs  207  times, which is  5.13265559137  percent.
mal occurs  185  times, which is  4.5871559633  percent.
noch occurs  175  times, which is  4.33920158691  percent.
als occurs  159  times, which is  3.94247458468  percent.
dass occurs  139  times, which is  3.44656583189  percent.
--------------------------------------------------------------------
Printing top 10 discontinuous connectives:
wenn  ...  auch occurs  16  times, which is  69.5652173913  percent.
von  ...  wegen occurs  2  times, which is  8.69565217391  percent.
entweder  ...  oder occurs  2  times, which is  8.69565217391  percent.
sowohl  ...  als auch occurs  1  times, which is  4.34782608696  percent.
weder  ...  noch occurs  1  times, which is  4.34782608696  percent.
auf  ...  hin occurs  1  times, which is  4.34782608696  percent.
teils  ...  teils occurs  0  times, which is  0.0  percent.
umso  ...  als occurs  0  times, which is  0.0  percent.
umso weniger  ...  als occurs  0  times, which is  0.0  percent.
zu  ...  als dass occurs  0  times, which is  0.0  percent.
--------------------------------------------------------------------
Printing top 10 ambiguous connectives:
und occurs  798  times, which is  21.0609659541  percent.
aber occurs  363  times, which is  9.58036421219  percent.
ja occurs  278  times, which is  7.33702823964  percent.
wie occurs  221  times, which is  5.83267352864  percent.
```

```
fr occurs  215  times, which is  5.67432040116  percent.
dann occurs  207  times, which is  5.46318289786  percent.
mal occurs  185  times, which is  4.88255476379  percent.
noch occurs  175  times, which is  4.61863288467  percent.
als occurs  159  times, which is  4.19635787807  percent.
dass occurs  139  times, which is  3.66851411982  percent.
--------------------------------------------------------------------
```

5 Known Problems and Future Work

- **Optimization** - the tool is quite slow, and could likely be improved through optimization or potentially the incorporation of existing tools or technologies. Using the `search()` and `findall()` functions from the BeautifulSoup package is expensive, as is the handling of large files.
- **Testing** - Due to memory and data constraints, we were unable to perform very thorough testing. Given the variety of connectives and diversity of Twitter data, this is necessary, and some errors can be seen in the data. A specific example is the handling of connectives in DiMLex that have both single and phrasal orthographies.
- **Additional disambiguation** - Schneider's work includes results for only a small fraction of the discourse markers included in DiMLex. While this includes many of the common discourse connectives, other ambiguous connectives remain. Additional research or other methods could be used to improve disambiguation.
- **Conversations** - The project currently focuses on single tweets, not pairs or conversations among Twitter users. Since tweets are very short texts, and many discourse connectives allow arguments over larger spans, this should be further expanded.

Acknowledgements

Thanks to Paul Ebermann for some suggestions on optimization and Wladamir Sidorenko for providing us with parallel POS-annotated tweet files.

References

[1] Angela Schneider. *Disambiguierung von Diskurskonnektoren im Deutschen* (Unpublished bachelor's thesis). Universität Potsdam.

[2] Angela Schneider & Manfred Stede. (2012). *Ambiguity in German connectives: A corpus study.* In: *Proceedings of KONVENS 2012 (Main track: poster presentations)*, 254-258.

[3] Anne Schiller, Simone Teufel, Christine Stöckelt & Christine Thielen. (1999). *Guidelines für das Tagung deutscher Textcorpora mit STTS (Kleines und großes Tagset).*

[4] Manfred Stede. (2002). *DiMLex: A lexical approach to discourse markers.* In: A. Lenci, V. Di Tomaso (eds.): *Exploring the Lexicon - Theory and Computation.* Alessandria (Italy): Edizioni dell'Orso, 2002.

[5] Uladzimir Sidarenka, Tatjana Scheffler, & Manfred Stede. (2013). *Rule based normalization of German Twitter messages.* In: *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology.*

[6] Wladimir Sidorenko. (20 August 2015). University of Potsdam staff home page. `http://www.ling.uni-potsdam.de/staff/sidorenko`