TweetGeo – A Tool for Collecting, Processing and Analysing Geo-encoded Linguistic Data

Nikola Ljubešić
Dept. of Knowledge Technologies
Jožef Stefan Institute
nikola.ljubesic@ijs.si

Tanja Samardžić
CorpusLab, URPP Language and Space
University of Zurich
tanja.samardzic@uzh.ch

Curdin Derungs
GISLab, URPP Language and Space
University of Zurich
curdin.derungs@geo.uzh.ch

Abstract

In this paper we present a newly developed tool that enables researchers interested in spatial variation of language to define a geographic perimeter of interest, collect data from the Twitter streaming API published in that perimeter, filter the obtained data by language and country, define and extract variables of interest and analyse the extracted variables by one spatial statistic and two spatial visualisations. We showcase the tool on the area and a selection of languages spoken in former Yugoslavia. By defining the perimeter, languages and a series of linguistic variables of interest we demonstrate the data collection, processing and analysis capabilities of the tool.

1 Introduction

Geographic distribution of linguistic features is traditionally studied in dialectology (regarding closely-related varieties) and in language typology (regarding different languages and language families), with the goal of identifying the patterns of language change. The potential for studying the geographic spread of linguistic features increased with the development of computer-mediated-communication (CMC). Short and long texts produced by the users of social media communication platforms constitute large samples of authentic language use that can be automatically retrieved and analysed to address a range of questions about human behavior, including spatial linguistic patterns analysed in this paper.

The social network Twitter made an especially important contribution to the development of new methods of collecting linguistic data from the Internet by allowing access to the content produced by their users through an API (application programming interface). One interesting feature of Twitter is that tweets are often associated with spatial information, explicitly (GPS coordinates) or implicitly (place names). The data collected from Twitter can either be used as a valuable complement to linguistic data already collected by traditional means or, if traditional data is not available, as a replacement. This opportunity, however, comes with considerable challenges. First, using the data collection interface requires technical skills that researchers interested in studying language variation usually cannot be expected to have. Second, once collected, the data often turns out to be noisy, difficult to annotate and unevenly distributed in space. Observing patterns therefore requires advanced processing methods, such as spatial statistics and geographic information science (GIScience).

In this paper, we present a tool set that combines computational linguistics and GIScience methods in order to facilitate the collection, visualisation and analysis of georeferenced tweets. Our major goal is to allow the wider linguistic research community access to data automatically collected from the Internet (Twitter data in this particular case). We provide a configurable tool set for speeding up the research process without limiting researchers in their choice of input data and analysis techniques. The user is, for instance, free to specify language(s), regions, and linguistic features of interest. Spatial analysis tools

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3412–3421, Osaka, Japan, December 11-17 2016.

can be applied to the extracted and annotated data for visualization and as a help in reasoning about the spatial patterns. In short, we present a tool intended to enable researchers with basic programming skills to perform advanced computational and spatial linguistic analysis.

Throughout the paper, we illustrate the functioning of the tool on an example data set collected for the territory of former Yugoslavia. We choose this region as a case where collecting new linguistic data is especially important. Due to the recent linguistic proliferation,[1] the current situation in this region provides an opportunity for all interested researchers to observe the impact of historical developments on language change. Collecting and analysing samples of computer-mediated-communication becomes particularly important since conducting large-scale linguistic surveys is impeded by the current political and economical situation.

2 Related Work

Computational analyses of the geographic distribution of linguistic features have a long tradition in dialectological research. The data for studying the spread of linguistic features are traditionally collected through questionnaires and field work: a number of potentially informative categories are selected and their realisations are elected from a number of informants selected to represent a linguistic variety in a particular area. Data collected in this way are then stored in databases that can be queried and used for different kinds of quantitative analyses (Nerbonne, 2009; Bauernschuster et al., 2014; Szmrecsanyi, 2012; Wieling et al., 2011). The knowledge about the distribution of linguistic features on the world-wide scale has recently become available in the form of databases. For instance, the data stored in a well-known typological database, WALS (Dryer and Haspelmath, 2013) is often used in large-scale computational studies of language universals (Dunn et al., 2011).

The trends in computational analysis of spatial linguistic data sets led to the development of specialised software such as GeoLing[2], a tool, written in Java, that enables researchers to visualise the spatial distribution of linguistic features using methods such as kernel density estimation (for smoothing data points representation on maps), factor analysis (for reducing the dimensionality) and clustering (for grouping similar areas). This tool requires a previously prepared data set, which can be collected using traditional methods.

User-generated content available on the Internet is used in computational linguistics mostly to study demographic characteristics of speakers based on the linguistic variety they use (Eisenstein et al., 2011; Nguyen et al., 2011; Danescu-Niculescu-Mizil et al., 2013), often including a geographic component (Doyle, 2014; Hovy and Johannsen, 2016). We concentrate here on the work where associated software was made available. Doyle (2014), proposes a method based on conditional probability to estimate the geographical distribution of linguistic features using Twitter. This method can be used to overcome the problem of uneven geographic distribution of collection points.[3] The software associated with this work is SeeTweet[4], a Python tool that uses the Twitter search API to collect tweets containing terms of interest, as well as base terms used for estimating the prior spatial frequency of tweets. The tool does not perform any visualisation of the collected data or any inference.

A recently developed web-based tool called Humboldt[5] (Hovy and Johannsen, 2016) provides a search interface that allows the user to query for lexical phenomena in five languages, and to get both statistical analysis and map representations of the results along two demographic factors: age and sex. This tool uses a data set previously collected by the authors from one source of online reviews of companies.

The tool that we propose in this article differs from the existing tools in its scope and flexibility. Previous tools are mostly focused either on data collection (SeeTweet) or analysis (GeoLing, Humboldt). We integrate these two components allowing the researchers to set up their own criteria both for collecting

[1] Following the war and the separation of SFR Yugoslavia's constitutive republics in the nineties, one of the official languages, Serbo-Croatian, was divided into four languages: Croatian, Bosnian, Montenegrin, and Serbian.

[2] `https://www.uni-ulm.de/en/mawi/geoling/home.html`.

[3] Note that the problem of uneven distribution does not concern traditional data collection methods, where balanced sampling (based on ZIP codes, for instance) is usually part of the design.

[4] `https://github.com/gabedoyle/seetweet`.

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

and analysing data sets depending on their hypotheses. With our tool, the user can define a wide range of potential features to be extracted from a large set of messages. Unlike SeeTweet, where data collection is limited to searching for specific terms, we provide the possibility for capturing the whole (available) data stream, storing all the messages produced in a given region during the collection time. The extraction of the features of interest is again controlled by the user who applies predefined generic functions on his linguistic description of specific phenomena, consisting of either regular expressions or a lexical resource. Regarding the analysis, our tool offers flexibility by allowing users to dynamically switch between spatial summary statistics, simple visualisations and more sophisticated analysis. Importantly, all analysis steps are independent from the underlying spatial distribution of the collected data. This is important since CMC geo-encoded data is known to be biased towards places with high population and sparse in rural regions (Hecht and Stephens, 2014).

In the following three sections we describe the tool and the research set-up that it supports. Each section describes one of the three main components of the tool. With an example data set we illustrate how each component is configured and what it gives as a result. The tool, accompanied with the exemplary dataset, is made available on GitHub[6].

3 Data Collection

The data collection component communicates with the Public Twitter Streaming API and stores the messages to a given location.

It is written in Python and relies on the *tweepy* Twitter API wrapper. In order to start data collection, the user needs to edit the configuration file by entering his Twitter API credentials (obtained from the Twitter Developer site), the project name (arbitrarily defined by the user) and the perimeter of interest (defined by the longitude and latitude bounds). Once the process is launched, a database is created and all the messages obtained from the Public Streaming API are stored. Messages not containing explicit longitude and latitude are discarded.[7]

Each of the retrieved objects is stored in an *sqlite* database as a BLOB structure. Parts of these structures, the `lang` and `screen_name` attributes, are also explicitly stored in the database outside the BLOB for reporting purposes. From these entries, the user can get a collection update at any time, specifying the number of tweets collected, the number of speakers (Twitter users) who published the tweets, and the head of the frequency distribution of tweets per speaker and tweets per the `lang` attribute value.

For our use case we started the data collection procedure in January 2016. For illustration purposes, we use the data collected up to July 2016, while continuing to run the collection process. We defined the perimeter over the countries of former SFR Yugoslavia, namely Slovenia, Croatia, Bosnia, Montenegro, Serbia and FYR Macedonia. During the 6 month period we collected 526,658 tweets from 50,783 speakers.

4 Data Processing

The data processing module is written in Python. Its two main functionalities are data filtering and extraction of linguistic variables.

4.1 Data Filtering

At this point, we provide three speaker-level (i.e. Twitter-user-level) filtering criteria:

- the minimum number of tweets published by a speaker,
- the most prominent language(s) used by a speaker, and
- the most prominent countries from which the speaker tweets.

[6] `https://github.com/clarinsi/tweetgeo`

[7] There are two ways of encoding spatial information on Twitter, explicit position (longitude, latitude), or a location (ranging from a town to a country). We discard the latter as in many areas locations are too general to be useful for spatial analysis.

While implementing the first and last filtering functionalities was quite straightforward (the country from which the tweet was sent can be obtained directly from the `Status` object), the language filtering functionality required additional engineering. Namely, although Twitter messages are tagged with the `lang` attribute, this attribute is known to be very unreliable, especially for the so called smaller languages. To filter only the messages in the language(s) of interest, we perform additional language identification on the level of speaker by applying the off-the-shelf language identification tool *langid.py*[8] (Lui and Baldwin, 2012) on a concatenation of all tweets of a speaker. Before performing language identification, we remove mentions, hashtags and URLs , as such elements were not seen in the language identification training data.

Only the tweets produced by the speakers that satisfy the language constraints are passed to the variable extraction module.

The language filtering criteria are set in a configuration file which is shared with the feature extraction process. In our use case we allowed three languages known to *langid.py*: Croatian, Bosnian and Serbian. While we were collecting data published in Slovenia and FYR Macedonia as well, in this analysis we are not interested in Slovene nor Macedonian data, but want to retain the speakers of the languages of interest from these countries. For this use case we did not define any restrictions regarding the minumum number of tweets per user, and we allowed tweets from our countries of interest and their neighbouring countries.

By running the speaker-level language identification over the fifty thousand Twitter users in our example collection, we have identified 3,854 speakers of the languages of interest. Given that only 7% of the total of collected speakers were identified as writing in the defined languages, we have performed an evaluation of the language identification output by manually checking 200 random entries. While the precision on this sample was 1.0 (16 out of 200 cases), recall was 0.89 as two users were not identified as speakers of the languages of interest. However, each of the two missed users actually produced just one tweet consisting of two words beside smileys and mentions, making the loss negligible.

While there are four times more English speakers than those of the languages of interest, there is a comparable number of Italian speakers and a smaller number of Russian, Spanish, Turkish and Slovene speakers.

4.2 Variable Extraction

Variables represent the user's linguistic features of interest. Our tool allows for a great flexibility in defining the variables, allowing the researchers to express their theoretical insight and creativity in formal description of the linguistic phenomena, putting more weight on individual linguistic insights than it is usually the case in quantitative approaches which tend to use aggregate linguistic data. Deeper exploration of linguistic features and their interactions is in line with the current trends in spatial linguistic research (Wieling and Nerbonne, 2015).

We showcase the variable extraction module on five nominal variables relevant for our area of interest. The first variable, *yat* (illustrated in Table 1), covers the Proto-Slavic vowel which has a different reflex in different dialects, having two levels, *e* for text containing forms of the Ekavian dialect (*dete*, 'child') and *je* for that containing forms of the Jekavian dialect (*dijete*). The second variable, *štošta* focuses on the variation in the interrogative pronoun *what*, with two levels, Standard Croatian *što* and Standard Serbian *šta*. The third variable *daje* covers two levels of variation in the interrogative clitics, *je li* prescribed in Standard Croatian, and *da li* allowed in the remaining varieties. The fourth variable, *month* covers a two-level lexical variation, where Croatian contains specific names for months (*siječanj*, *veljača*...) encoded via variable level *hr*, while the remaining varieties use international ones (*januar*, *februar*...) encoded via variable level *int* (see Table 1 for some examples). The fifth variable, *rdrop* (also given in Table 1), covers the frequent drop of the ending *r* in Standard Serbian like *jučer* (encoded with level *r*) vs. *juče* (encoded with level *nor*).

Our variable extraction component is defined in the configuration file mentioned in the previous subsection in the form of four lists of functions.

[8] `https://github.com/saffsd/langid.py`

yat		month		rdrop	
Word	Value	Word	Value	Word	Value
dječak	je	juni	int	juče	nor
dečak	e	lipanj	hr	jučer	r
dječaka	je	august	int	naveče	nor
dečaka	e	kolovoz	hr	navečer	r

Table 1: A sample of a feature extraction lexicons; each column represent one file used by the extraction function.

The first list of functions operates on the `Status` object of the *tweepy* module, enabling extraction of metadata such as the number of retweets, posting time, whether the tweet is a reply to another tweet etc. The formalism requires for the user to define the location of the metadata in the *tweepy* `Status` object, like `user.screen_name` for the speaker's screen name or `favorited_count` for the number of times a status was favorited. The user can additionally define a function to be applied on the metadata value, such as extracting the posting year from the posting time, like `lambda x:str(x.year)`.

The remaining three function lists operate on the text of the tweet. While the second list of functions operates on the original text of the tweet, the third list of functions operates on the lowercased text, and the fourth one on the normalised text of the tweet.

The normalisation process can be defined by the user and covers, in our use case, removal of repeating characters, generalising spaces and removal of diacritics.

The choice of the text representation level from which the variable will be extracted depends on how important the literal representation of the writing is for extracting a particular variable. For instance, when idetifying the *što* pronoun, we use second level text representation – lowercased text. We want to easily take into account titlecased or uppercased versions of the pronoun, but do not want for diacritics to be removed as the form *što* clashes with the numeral *one hundred*. On the other side, when identifying names of months, we use the third level of text representation, covering with the form *ozujak* various forms like *Ožujak*, *ozujak* or *ožuJAAAAAK*.

The functions that can be run on any of the three mentioned text representations are divided into two types: the `lexicon_choice` and the `regex_choice` function.

4.2.1 Lexicon Choice

In the functions of the `lexicon_choice` type, the desired feature to be extracted is encoded by the user in the form of a lexicon file, where each line consists of a (word, value) pair. An example of such a lexicon in our use case is the lexicon of words containing the already mentioned Proto-Slavic vowel *yat*, as illustrated in the first column of the Table 1.

The *e* reflex is characteristic in the eastern variants (mostly Serbian), while the *je* reflex is found more to the western side of our target perimeter (Croatian, Bosnian, Montenegrin).

The lexicon illustrated in Table 1 was automatically generated from the Croatian and Serbian inflectional morphological lexicons hrLex and srLex (Ljubešić et al., 2016) by searching for pairs of words having the same morphosyntactic description and the word forms identical except the transformations (*ije* vs. *e*) or (*je* vs. *e*)), and both word forms having just one possible canonical form (lemma).

The `lexicon_choice` function iterates through the tokens of each of the collected tweets. If any of the tokens matches any word in the lexicon, the variable value associated with the word is added to the set of potential values. If, at the end of the tweet, there is only one value in the set of potential values, the function assigns that value to the tweet. In all other cases (no coverage, multiple values) the function returns the `NA` value.

The tokenisation function can be also modified by the user. It currently considers tokens to be hashtags, mentions, URLs or greedy alphanumeric sequences.

Similar lexicons can be specified by the user to extract any lexical variation features. Such lexicons can be written by hand or extracted automatically from other resources such in our case. Once they are stored in a location required by the function, they can be used for extraction.

In our use case, we have extracted four such lexicons, illustrated in Table 1. We have chosen these variables based on their varied use, as discussed in comparisons of the language varieties (Meša, 2011). Detailed documentation of the extracted variables is available in the tool documentation.

4.2.2 Regex Choice

The `regex_choice` function operates similarly to the `lexicon_choice` function, it just does not rely on a lexicon, but a list of pairs of regular expressions and variable values. If a regular expression is applicable to the text of a tweet, the corresponding variable value is added to the set of potential values. The decision which value should be returned is identical as in the `lexicon_choice` function. An example of such a function in our use case is the variation of the particles *je li* and *da li*, each being covered by the corresponding regular expressions `"\bje li\b"` and `"\bda li\b"`. These regular expressions are applied on the third level of text representation, namely normalised text.

5 Data Analysis

The analysis module is written in R, an open-source programming language which incorporates a wealth of packages for spatial data handling. At this level of the tool development we decided to limit ourselves to three functionalities: point visualisation, spatial trend detection and the identification of dominant regions per variable level. In the remainder of this section, each of the three functionalities is illustrated.

Here we stress one more time that in this analysis we only consider tweets associated with explicit geolocation, which is only given for some 1-3% of all Twitter messages (Leetaru et al., 2013). We do not attach geolocation to unlocated tweets by, for instance, using the place of domicile of the user location prediction procedures, as it has been shown that such procedures can lead to wrong assumptions (Hahmann et al., 2014).

5.1 Point Visualisation

The point visualisation allows to gain an initial impression of the spatial distribution of all levels of a linguistic feature. It can thus be considered a visual analytics tool (Andrienko et al., 2010).

For visualisation we use the leaflet framework [9], which allows to dynamically change the spatial focus by zooming and panning. This functionality proves to be vital for representing tweets as spatial points, due to the uneven spatial distribution discussed above. On small scales (e.g. country level) tweets are cluttered in populated places and it is thus often difficult to identify the exact distribution of feature levels without having the option of dynamically changing the scale and extent of the map.

Additionally, leaflet allows to activate an HTML popup option, which we use to allow access to the text content of each tweet through mouseclick. The user can for instance iterate through a subsample of the data and thus gain an impression on the data quality in terms of the spatial precision or linguistic variable extraction.

This functionality is also intended to be used alternately with the variable extraction module. Namely, besides analysing the output of the variable extraction process in pure text format, it is often easier to analyse the extracted variables in space and therefore get a faster insight in potential problems in the variable extraction process.

Examples of point visualisation on the *yat* and *štošta* variables can be seen on the left side of Figure 1 and Figure 2.

5.2 Spatial Trend Detection

Spatial linguistic analysis is often concerned with first-order effects, such as distributional patterns in the data (Diggle, 2014). With the spatial trend detection tool we intended to go one step further and introduce a simple measure that allows to quantify the spatial dependency in the data, often referred to as spatial autocorrelation or second-order effect. The quantification of spatial autocorrelation for continuous variables (e.g. temperature) is well established and measures such as *Moran's I* can be used (Moran, 1950). Quantifying spatial autocorrelation in nominal data, which we deal with in this paper, is

[9] `http://leafletjs.com/`

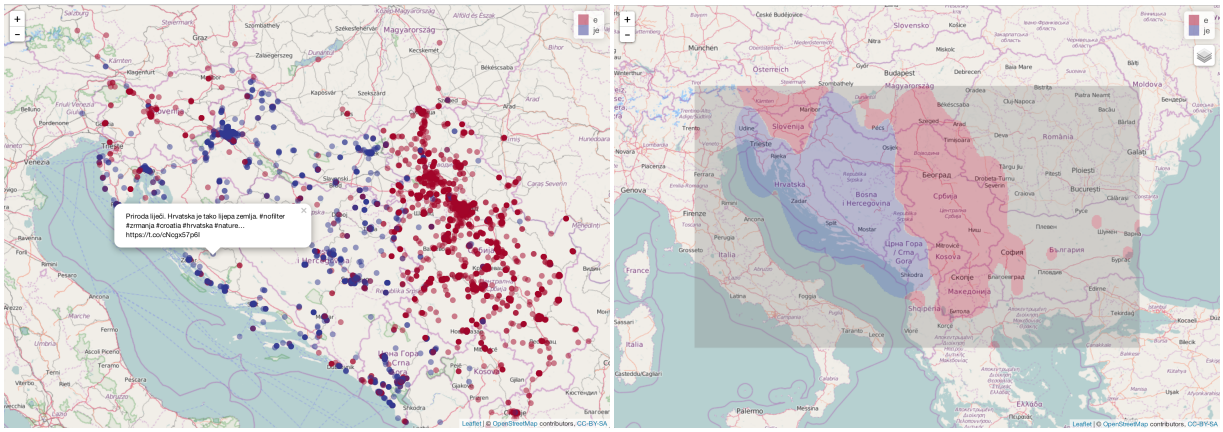

Figure 1: Result of point visualisation (left) and dominance map (right) for the variable *yat*

variable	level	spatial trend	frequency
yat	e	0.457	3298
	je	0.812	1702
stosta	što	1.079	1097
	šta	0.961	2205
daje	dali	0.902	497
	jeli	1.684	58
month	hr	1.252	16
	int	0.906	416
rdrop	r	0.402	79
	nor	0.713	619

Table 2: Statistical description of each variable and level

slightly less common. We compare the spatial distances as computed between all tweets of one linguistic feature (expected distances) with the distances as calculated for each feature level separately (observed distances). Aggregating these two sets of distances into what we call a *relative distance measure* allows us to distinguish feature levels that are spatially clustered (observed distance $<$ expected distance) from levels that are scattered in space (observed distance $>$ expected distance).

The results of the spatial trend detection applied to our five features is given in Table 2. We can observe that two variables having a strong spatial trend (low value equals strong trend), namely *yat* and *rdrop*.

In the *yat* variable the *e* level shows a higher spatial trend than the *ije* level, mostly due to the fact that the use of Ekavian is focused around Belgrade while the use of Jekavian is much more scattered around. This trend can also (partially) be observed in the point visualisation in Figure 1.

In the *rdrop* variable, the *r* level shows a much stronger spatial trend, which goes back to the fact that these variants are mostly used in Croatia only while in the remainder of the region the *nor* variants are used.

A somewhat surprising spatial trend is that of the *month* variable for which we would expect to have a strong spatial trend especially the *hr* level as Croatian month names are used in Croatia only. This result can be followed back to a low observation frequency of the variable in general, especially of the *hr* level. This result shows that, as most measures, the spatial trend heuristic is prone to outliers for small sample sizes.

The remaining two variables, *štošta* and *daje* show an expected weak spatial trend (higher is weaker) as these variants are used intermittently in the whole area of interest.

As shown with these examples, the spatial trend detection tool serves as a simple heuristic for deciding which linguistic features bear the potential of segregating space into larger linguistic areas. Ideally,

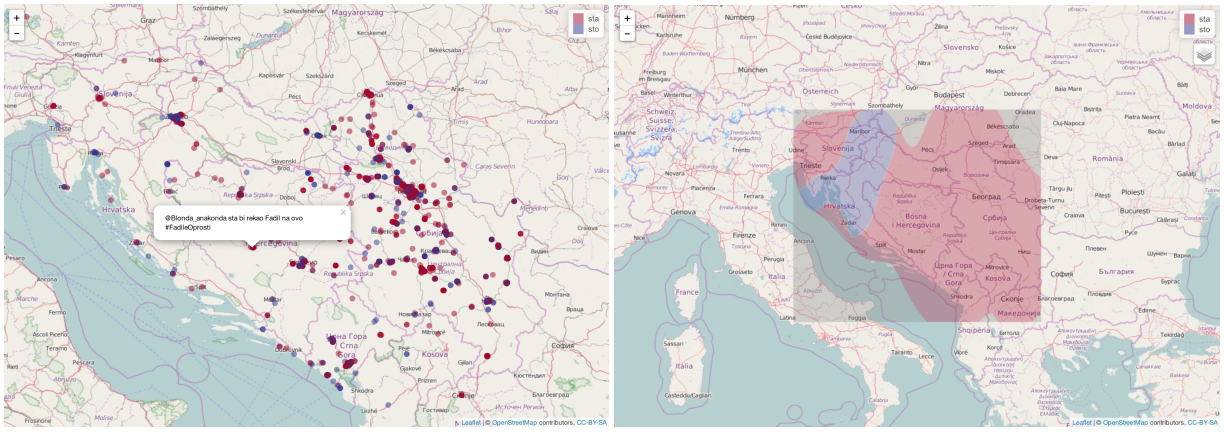

Figure 2: Result of point visualisation (left) and dominance map (right) for the variable *štošta*

candidate features will be passed forward to the dominance map tool, described in the next section.

5.3 Dominance Maps

The dominance map functionality provides the means for calculating continuous surfaces from point observations, in our case georeferenced tweets. Surfaces are calculated for each feature level separately, using kernel density estimation (KDE), a well established method for representing point observations as density surfaces. The local value of a density surface represents the number of observations of the respective feature level proximate to this location. A kernel function is applied for smoothing the signal and to thus account for local noise. The application of KDE to linguistic data is well represented in literature, e.g. (Bart et al., 2013). After computing density surfaces for each feature level individually, local intensities are compared and only the level with maximum local intensity is preserved and mapped as the dominant level. Hence, the dominance map function visually represents linguistic areas *dominated* by individual feature levels.

The two variables presented via point visualisation on the left side of Figure 1 and Figure 2 have their dominance maps depicted on the right side of the respective figures. While for the *yat* variable the point visualisation was already informative, due to a strong spatial trend as reported in Table 2, for the *štošta* variable, showing a weaker spatial trend, the visual identification of regionally dominant levels turns out to be considerably difficult. Therefore the dominance map comes in very handy, showing that only in central Croatia the *što* level is dominant while in the rest of the study area *šta* dominates.

For both variables the dominance map shows that the value dominant in Bosnia and Serbia is also dominant in Slovenia. This is due to large Bosnian and Serbian communities living in this country.

The results of the analysis of the *štošta* variable will benefit from more extensive data collection as the point visualisation shows that the areas of Croatia and Bosnia are only sparsely covered. We would therefore like to emphasize the preliminary nature of these results.

6 Conclusion and Future Work

In this paper, we have presented a configurable, flexible tool set for working with geo-encoded linguistic data automatically collected from Twitter. We have demonstrated through a use case how the tool facilitates to monitor language use in a region of interest. We have extracted a sample of five features with varied linguistic properties (phonetic, lexical, syntactic) and analysed their spatial distribution using spatial methods of varied complexity.

The results of our initial analyses indicate that the flexibility offered by our tool is important for gaining important insights into the data: higher level analyses and visualisation can reveal patterns not visible in a simpler point visualisation (as in the case of the *štošta* variable). On the other hand, point visualisation can serve as a good tool for manual checkups of the reliability of both the extracted data and higher-level analyses. With the possibility to obtain different representations quickly and in a relatively simple

way, researchers can use our tool to address important questions regarding geographical distributions of linguistic features. As the tool is written in popular languages, it is also easy to extend.

In future work, we will use the presented tool to perform further analyses of the language use in the region addressed in our initial study. We will address some of the key points in the ongoing debate about the differences between Bosnian, Croatian, Montenegrin and Serbian and the potential process of linguistic separation.

Furthermore we will continue extending the presented TweetGeo tool with additional analysis capabilities, as well as work on merging the tool with previously developed tools – TweetCat[10] (Ljubešić et al., 2014) which focuses on data acquisition through the Twitter Search API, and TweetPub[11] which is meant for preparing linguistically annotated Twitter collections for publishing while following the Twitter Developer agreement – into a unified toolkit for gathering, analysing and redistributing Twitter data.

While the presented tool is designed for spatial linguistic analysis, we would argue that it is also suited for studying the spatial distribution of other phenomena that can be studied using Twitter and associated metadata. Examples are demographic characteristics, relations between the speakers or particular ways of using the social network. We would therefore argue that without major changes, our tool could be applied to the broader context of the humanities.

Acknowledgements

We would like to thank our collaborator Dolores Batinić for her help in the linguistic part of our work. The work presented in this paper was supported by the URPP 'Language and Space' individual grant and by the Swiss National Science Foundation grant No. 160501.

References

Gennady Andrienko, Natalia Andrienko, Urska Demsar, Doris Dransch, Jason Dykes, Sara Irina Fabrikant, Mikael Jern, Menno-Jan Kraak, Heidrun Schumann, and Christian Tominski. 2010. Space, time and visual analytics. *International Journal of Geographical Information Science*, 24(10):1577–1600.

Gabriela Bart, Robert Weibel, Pius Sibler, and Elvira Glaser. 2013. Analysis of Swiss German syntactic variants using spatial statistics. *Álvarez Pérez, Xosé Afonso, Ernestina Carrilho & Catarina Magro (red.). Current Approaches to Limits and Areas in Dialectology. Newcastle upon Tyne: Cambridge Scholars Publishing*.

Stefan Bauernschuster, Oliver Falck, Stephan Heblich, Jens Suedekum, and Alfred Lameli. 2014. Why are educated and risk-loving persons more mobile across regions? *Journal of Economic Behavior & Organization*, 98:56 – 69.

Cristian Danescu-Niculescu-Mizil, Robert West, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. No country for old members: User lifecycle and linguistic change in online communities. In *Proceedings of WWW*.

Peter J . Diggle. 2014. *Statistical Analysis of Spatial and Spatio-Temporal Point Patterns, Third Edition*. Chapman and Hall/CRC 2013.

Gabriel Doyle. 2014. Mapping dialectal variation by querying social media. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 98–106, Gothenburg, Sweden, April. Association for Computational Linguistics.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Michael Dunn, Simon J. Greenhill, Stephen C. Levinson, and Russell D. Gray. 2011. Evolved structure of language shows lineage-specific trends in word-order universals. *Nature*, pages 79–82.

Jacob Eisenstein, Noah A. Smith, and Eric P. Xing. 2011. Discovering sociolinguistic associations with structured sparsity. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1365–1374, Portland, Oregon, USA, June. Association for Computational Linguistics.

[10] https://github.com/clarinsi/tweetcat
[11] https://github.com/clarinsi/tweetpub

Stefan Hahmann, Ross S Purves, and Dirk Burghardt. 2014. Twitter location (sometimes) matters: Exploring the relationship between georeferenced tweet content and nearby feature classes. *Journal of Spatial Information Science*, 2014(9):1–36.

Brent Hecht and Monica Stephens. 2014. A tale of cities: Urban biases in volunteered geographic information. *ICWSM*, 14:197–205.

Dirk Hovy and Anders Johannsen. 2016. Exploring language variation across Europe - a web-based tool for computational sociolinguistics. In Nicoletta Calzolari et al., editor, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may. European Language Resources Association (ELRA).

Kalev Leetaru, Shaowen Wang, Guofeng Cao, Anand Padmanabhan, and Eric Shook. 2013. Mapping the global twitter heartbeat: The geography of Twitter. *First Monday*, 18(5).

Nikola Ljubešić, Darja Fišer, and Tomaž Erjavec. 2014. TweetCaT: a Tool for Building Twitter Corpora of Smaller Languages. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Nikola Ljubešić, Filip Klubička, Željko Agić, and Ivo-Pavao Jazbec. 2016. New Inflectional Lexicons and Training Corpora for Improved Morphosyntactic Annotation of Croatian and Serbian. In Nicoletta Calzolari et al., editor, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may. European Language Resources Association (ELRA).

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *ACL (System Demonstrations)*, pages 25–30.

Emira Mešanović Meša. 2011. *Kontrasivna analizabosanskog, hrvatskog i srpskog jezika u zakonima Federacije Bosne i Hercegovine*. Slavistički komitet, Sarajevo.

P. A. P. Moran. 1950. Notes on continuous stochastic phenomena. *Biometrika*, 37(1/2):17–23.

John Nerbonne. 2009. Data-driven dialectology. *Language and Linguistics Compass*, page 175–198.

Dong Nguyen, Noah A. Smith, and Carolyn P. Rosé. 2011. Author age prediction from text using linear regression. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 115–123, Portland, OR, USA, June. Association for Computational Linguistics.

Benedikt Szmrecsanyi. 2012. Geography is overrated. In Sandra Hansen, Christian Schwarz, Philipp Stoeckle, and Tobias Streck, editors, *Dialectological and folk dialectological concepts of space*, pages 215–231, Berlin. de Gruyter.

Martijn Wieling and John Nerbonne. 2015. Advances in dialectometry. *Annual Review of Linguistics*, pages 243–264.

Martijn Wieling, John Nerbonne, and R. Harald Baayen. 2011. Quantitative social dialectology: Explaining linguistic variation geographically and socially. *PLoS ONE*, 6(9):1–14, 09.