Geographical Visualization of Search Results in Historical Corpora

Florian Petran
Ruhr-Universität Bochum
petran@linguistics.rub.de

Abstract

We present ANNISVis, a webapp for comparative visualization of geographical distribution of linguistic data, as well as a sample deployment for a corpus of Middle High German texts. Unlike existing geographical visualization solutions, which work with pre-existing data sets, or are bound to specific corpora, ANNISVis allows the user to formulate multiple ad-hoc queries and visualizes them on a map, and it can be configured for any corpus that can be imported into ANNIS. This enables explorative queries of the quantitative aspects of a corpus with geographical features. The tool will be made available to download in open source.

1 Introduction

Work on visualizing language corpora usually focuses more or less on specific phenomena, such as term relations or semantic structure of a text (such as (Cao et al., 2010) or (Fortuna et al., 2005)). Quantitative relations are represented with techniques from the data sciences, and the corpus is largely treated as an atomic entity — it is visualized as a whole, and not in parts. This is a necessity, first due to the broad and varied nature of linguistic phenomena present in annotated corpora, and due to the irrelevance of geographical distribution in most synchronic corpora. So in relation to corpora, visualization often means picturing complex annotations such as trees, or dependency relations. This type of visualization focuses on qualitative work.

Conversely, geographic visualization is usually done for the social sciences, or in historical studies. In those disciplines, queries are usually not formulated ad-hoc, but carefully prepared, since the data is harder to come by. So data is specifically gathered for a certain research question, and it is gathered to answer that question explicitly and only. Data sets are therefore less rich than linguistic corpora typically are. The tools reflect that in that they usually require the user to import query results generated elsewhere.

However, in corpora of dialects, or diachronic corpora, the geographical distribution can be a central, or at least very important feature. We present a solution for geographical visualization of arbitrarily complex, ad-hoc searches. The visualization itself is fairly simple — circles on a map, scaled for match count. However, the ability to combine it with very different type of corpora, and with user queries makes it a powerful tool for historical and dialectal research.

Section 2 below will discuss two tools that are partially similar in functionality. Section 3 explains the implementation of the application in technical detail (3.1), and the user interface (3.2). Section 4 shows our sample deployment. In Subsection 4.1 we give some brief details on the corpus used for the sample deployment, and Subsection 4.2 demonstrates the functionality on a short example. Finally, section 5 will discuss plans for further development.

2 Related Work

ANNIS (Krause and Zeldes, 2014) is a search architecture as well as an approach to the visualization of different annotation levels in corpora that is applicable for a vast array of linguistic corpora. It is realized as a web application with a browser interface as well as a webservice. Queries can be formulated in a

Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH), pages 94–100, Osaka, Japan, December 11-17 2016.

specialized language (ANNIS Query Language), that allows specification of annotations from sub-token to sentence levels, and simple or complex interactions between them. The ANNIS web interface present matches for the query in a KWIC-like format, where each match is shown with a context window to the left and right. This facilitates qualitative research by browsing the results, but for quantitative approaches, the results have to be exported for use in third party tools.

GeoBrowser[1] is a data sheet editor and geographical visualization tool that was developed for historical, or social sciences use. Data points are represented with scaled markers on a map overlay. The tool offers a range of historical maps to choose from by default. Data has to be imported in a spread sheet format, or entered in an online tool. The tool will then automatically retrieve the geolocations, place the markers accordingly, and, if applicable, cluster them according to the zoom level. If a timeline can be construed from the data, the tool will also allow to limit the visualization to a sub period, or run an animation of the timeline.

While GeoBrowser offers comprehensive options for geographical visualization, it has two main drawbacks for our purposes. First, the process of generating results and then importing them is too slow and cumbersome to allow for explorative visualization of ad-hoc queries. The process of exporting and importing implicitly assumes that the user has an exact plan of the phenomena he wants to visualize. And second, GeoBrowser is closed source with a centralized server architecture, while our tool will be made available to download as well.

Geographical visualization has a long tradition in dialectology that goes back to the late 19th century founding of the *Deutscher Sprachatlas* by Georg Wenker (1888). He had compiled a list of example sentences that were translated into local dialects by volunteers. The *Atlas* then mapped the geographic boundaries between the various pronounciation features (isoglosses) drawn from the example sentences. The mapping of isoglosses remained the dominant approach in dialect mapping for most of the 20th century. From 2001 on, the publishers of the *Atlas* have started to digitize the maps making them available online[2], but the material remains more or less static.

There is ongoing discussion on effective use of computer generated maps (Schmidt and Auer, 2011). Early work on this area focuses on combining digitized maps of isoglosses in semi-transparent overlays, allowing comparative view of various different maps. In the later 20th century, the approach of dialectometry used quantitative approaches and similarity calculations to visualize dialectal differences. Traditional means of quantitative mapping in cartography are choropleth maps, and isarithmic, or contour maps. Coropleth maps show variation in aggregated data by different color shades over predefined regions. Contour maps show regions with the same value as line-bounded areas on a map. Both approaches have been used for mapping of aggregated dialect data.

Similarities between pronounciations can be further visualized with beam maps, which show similarities with lines between sites – the darker the line, the more similar the data (Goebl, 1984). This relies on relatively simple similarity calculations, if the aggregate data is clustered, those clusters can also be visualized. For example, the composite cluster map approach (Kleiweg et al., 2004) tiles the map into polygons around collection sites, and then shades the polygon borders according to their distance in the cluster dendrogram.

Compared to these approaches, our visualization is fairly simplistic, and it is certainly not suitable for every research question. Its main advantage, however, is its flexibility, the ability to apply it to any corpus that can be imported into ANNIS and is annotated with some geographic features.

3 Application

3.1 Implementation

ANNISVis builds on an existing installation of the search tool ANNIS. ANNIS works as a client-server app itself, with a Java applet frontend and a Java server backend. It is highly configurable and able to import a large number of formats for linguistic corpora (Krause and Zeldes, 2014). A server application

[1] `http://geobrowser.de.dariah.eu/`
[2] `https://regionalsprache.de/`

in Python over WSGI queries the ANNIS webservice for a complete list of documents from the corpus. A document is the smallest sub-division of a corpus in ANNIS.

The server then looks for specific keywords in the meta annotations of the corpus (i.e. annotations that relate to an entire document) that denote a place of origin. The keywords are obviously corpus dependent, so they can be configured at deployment time of the application. On first launch, the coordinates for each place name are looked up using the Google geocoding API via the GeoPy library[3]. A fallback meta key for the location specification can be provided as well.

Since looking up the places for a corpus with a lot of documents can take a long time, the server caches the geolocations it looked up in a JSON file. In addition to the API lookup, the administrator of the server has the opportunity to supply a file with predefined locations. This serves as an override for cases the Google API gets wrong, such as historical dialect areas, or places that do not exist any more. In cases where multiple documents have the same location, they are combined for the purposes of displaying search results on the map. In this paper, we will refer to these combinations of document set and geolocation as location.

The text list is returned by the server webapp upon page loading, the rest of the functionality is realized client side in JavaScript (using JQuery[4] and the Google Maps API[5]) over XML requests to the ANNIS service. Each datum presented in the UI is in its own container element, so that the presentation can easily be adapted with CSS.

3.2 User interface

The search box allows the user to formulate any query using ANNIS Query Language (see Sec. 4.2 for a brief explanation). The ANNIS web service is then queried for the counts for each document set at each location separately, and the results are displayed as circles at their geolocation on the map as a result marker with a scaled size relative to the number of search results. The map is automatically centered to show all locations that are present in the corpus that has been configured.

Two techniques for scaling are available: global and local. With global scaling, the location with the highest number of search results receives the largest circle, and the other locations are scaled accordingly. It will be most useful for a corpus of roughly evenly sized documents. Local scaling adjusts each result marker according to the number of tokens in the document. It can be a way of dealing with a corpus that is somewhat unevenly distributed (as historical corpora often are due to limited text availability). However, not all phenomena are suitable for local scaling with this method. Low frequency phenomena make up a very small percentage of their documents, so they will result in very small markers that are not easily visible on the map. Both methods of scaling can therefore be selected by the user before submitting a search request.

Result markers have info popups that can be configured to show different information on that specific results. Currently, it shows a basic description of the query results in general, and the result count. Each query gets assigned a color with uniformly random distribution over red, green, and blue values. While predefined colors would look better, they are not feasible in a system where a user can make an arbitrary number of queries. The random distribution ensures that it is fairly unlikely that two query colors are too close to be distinguishable.

Multiple queries may show results in the same places, and the result markers of close by places may overlap or obscure each other due to the scaling. ANNISVis offers several ways of dealing with this. When the user clicks on markers that overlap, they will move apart ("spiderfy") with lines indicating their original locations (see Fig. 1). Furthermore, the user has options to hide markers, or to fore- or background markers. Operations can be performed through the query context menu, in which case they apply for all markers, or on the marker list accessible through the text list, in which case they apply only to the selected marker.

[3] https://github.com/geopy/geopy/
[4] https://jquery.com/
[5] https://developers.google.com/maps/

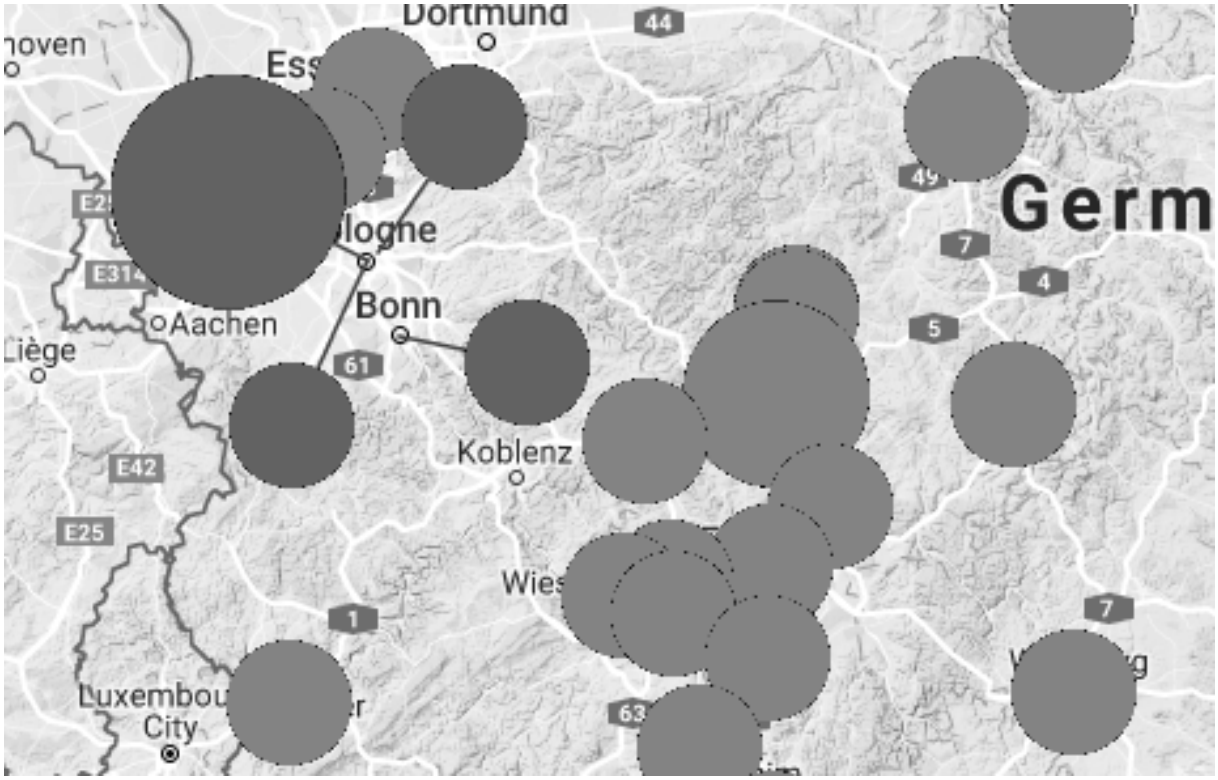

Figure 1: Part of a visualization with spiderfied markers. The spiderfied markers are in a slightly shaded version of the query color.

4 Sample Deployment

4.1 Language Data

The example below and our sample deployment uses the reference corpus Middle High German (REM), an annotated corpus of texts from about 1050 to 1350 AD (Petran et al., submitted). It combines and builds upon existing projects for historical corpora dating as far back as the 1980s. Some of the texts are fragments of only a few tokens, others are up to 100,000 tokens long. All texts are annotated with token level annotations (such as POS, morphology), and sub-token annotations for tokenization changes. The texts further have meta-annotation of the written dialect ("Schreibsprache") in different granularities, and some texts have their provenience annotated. The corpus aims to be largely balanced with regards to dialect, as far as the sources permit. This makes it very suitable to demonstrate the type of visualization our application can provide. However, especially the early part of the time period covered it was not possible to find enough texts for all dialects, which can lead to a skewed distribution of data, making it very suitable to

The meta key for the location lookup was configured to the place where the document was written. Unfortunately, this is not annotated for all documents. A number of factors come into play localizing a medieval document, including the place where a manuscript was written, where a scribe was born, where he was trained, and the adressee of a document. The obvious choice would be to use the dialect area annotation as fallback, however, those areas do not correspond to modern geographic entities, so no geocoding API recognizes them — they have to be manually geocoded. For the results shown in the screenshots below, we instead used the library where the manuscript is kept today as a placeholder. The library is not a valid localization by any of the standards outlined above, and the screenshots can therefore only serve as a proof of concept type demonstration. The real sample deployment will use manually geocoded dialect areas.

The geocoding worked well for the vast majority of the documents, but a few had to be manually corrected, such as St. Paul's abbey in Lavanttal (in Austria) which the API thought identical to St. Paul,

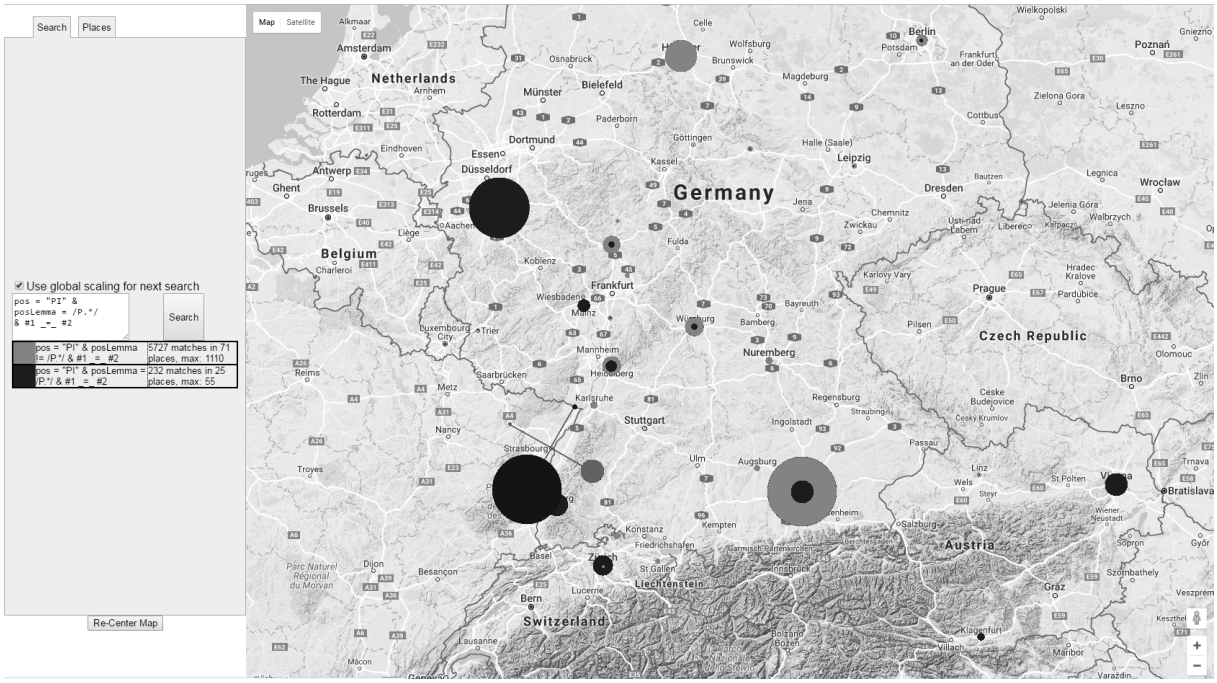

Figure 2: Visualization of two queries for pronouns

Minnesota.

A special feature of the corpus we used is its fine grained POS annotation (Dipper et al., 2013), which discerns the POS of the type (`posLemma`) and the POS of the token in its present usage (`pos`). For example, a prototypical adjective would be tagged ADJ as `posLemma`, and ADJA (attributive adjective) for `pos` if used in an attributive function (as opposed to an adverbial function for example). This enables a user to, among other things, study language change, as we will see below.

4.2 Example

An example user could be interested in studying pronouns in historical German dialects. Figure 2 shows such queries in our tool. The input box on the left shows the last user entered query in ANNIS Query Language (AQL). We will now briefly explain its syntax as far as it relates to this example[6].

Basic building blocks of AQL are search *nodes*, and interactions between the nodes. A search node is any query for annotations or annotation values. For example, a query for `pos` will return all elements with that annotation, while a query for `pos='PI'` will only return those that were annotated with PI (indefinite pronoun). `posLemma!=/P.*/` is a query using regular expressions that returns all elements where `posLemma` does not begin with `P` — that is, tokens that are not, according to their lemma, a pronoun.

As an example, consider (1). The token *man* — glossed as "one" here — is used as indefinite pronoun (PI), but the lemma originally means "man," a nomen appellativum (NA), and it can be used in this capacity in other places. This shows the language change from a single lemma *man* that can be used both as NA and PI to the modern situation where the PI is *man* and the NA is *Mann*.

(1)

	do	*man*	*daz*	*kint*	*befniden*	*fcolte*
`pos`	KOUS	PI	DDART	NA	VVINF	VMFIN
`posLemma`	KO	NA	DD	NA	VV	VM
	when	one	the	child	circumcise	should

"when the child should be circumcised"

As a counterexample, consider (2). The token *niemen* is used as PI, and the type is also already a PI.

[6]for detailed documentation see `http://corpus-tools.org/annis/aql.html`

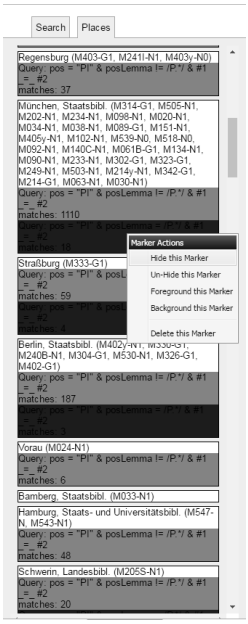

Figure 3: An entry in the text list

(2)

	ímo	*fcol*	*niemen*	*gelôbe*
`pos`	PPER	VMINF	PI	VVINF
`posLemma`	PPER	VMINF	PI	VV
	him	should	nobody	believe

"no one should believe him"

The search statement can refer to nodes by index in order of specification, so the `posLemma!=/P.*/` in Figure 2 is referred to as `#2`. A number of operators can specify how the annotations have to interact. In this example, `#1_=_#2` specify that both search conditions must apply to the same position.

In simpler words, we search for tokens that are used as indefinite pronouns, but are not originally pronouns in the first (light grey) query, and indefinite pronouns that are originally (according to their lemma) pronouns in the second (dark grey) query.

On the map (Figure 3), we can clearly see a distribution where western regions seem to favor original pronouns, while texts located in Bavaria and further north have a lot of pronouns that are not originally pronouns. It also shows how we spiderfied the markers around Strasbourg so we can tell apart the individual, overlapping results. The exact number of matches is displayed in the info popup for each location, which is not shown in the example, but also in the entry for each individual result in the location list tab (see Figure 3). The location list also offers marker actions (discussed in Section 3.2) as a context menu. The text list entries also give the match counts, to enable comparative evaluation of the query results on a location basis.

5 Future Work

In its current form, the system does not consider the diachronicity of the data at all. For future iterations, tieing the markers to a slider for the temporal axis, similar to what GeoBrowser offers, will be implemented.

Both local and global scaling are not always optimal, due to the reasons explained in Sec. 3 above. For future development we will look into alternative ways of scaling marker size, and possibilities of combining scaling for text size and scaling for global maxima.

Finally, we would like to explore the possibilities of searches based on annotation values, where the system can rank their relevance for each document with a measure like tf-idf, and display the most relevant values on their map locations.

Resources

The following resources are provided with this paper.

1. The source code of the application described in this paper. It can be found at `https://github.com/fpetran/annisvis`
2. A sample deployment using the reference corpus Middle High German (REM). It can be found at `http://www.linguistics.rub.de/annisvis`

Acknowledgements

My work has been funded by the Deutsche Forschungsgemeinschaft (DFG) grant DI 1558/1. I thank the anonymous reviewers for their helpful comments and suggestions.

References

Nan Cao, Jimeng Sun, Yu-Ru Lin, David Gotz, Shixia Liu, and Huamin Qu. 2010. Facetatlas: Multifaceted visualization for rich text corpora. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1172–1181.

Stefanie Dipper, Karin Donhauser, Thomas Klein, Sonja Linde, Stefan Müller, and Klaus-Peter Wegera. 2013. HiTS: ein Tagset für historische Sprachstufen des Deutschen. *Journal for Language Technology and Computational Linguistics, Special Issue*, 28(1):85–137.

Blaz Fortuna, Marko Grobelnik, and Dunja Mladenic. 2005. Visualization of text document corpus. *Informatica*, 29(4).

Hans Goebl. 1984. *Dialektometrische Studien: Anhand italoromanischer, rätoromanischer und galloromanischer Sprachmaterialien aus AIS und ALF*. Niemeyer, Tübingen.

Peter Kleiweg, John Nerbonne, and Leonie Bosveld. 2004. Geographic projection of cluster composites. In Alan Blackwell, Kim Marriott, and Atsushi Shimojima, editors, *Diagrammatic Representation and Inference*, pages 392–394. Springer, Berlin.

Thomas Krause and Amir Zeldes. 2014. ANNIS3: A new architecture for generic corpus query and visualization. *Digital Scholarship in the Humanities*.

Florian Petran, Thomas Klein, Stefanie Dipper, and Marcel Bollmann. submitted. REM: A reference corpus of Middle High German — corpus compilation, annotation, and access. *Journal for Language Technology and Computational Linguistics*.

Jürgen Erich Schmidt and Peter Auer, editors. 2011. *Language and Space: an international handbook of linguistic variation*. de Gruyter, Berlin, Boston.