Geolocation and Named Entity Recognition in Ancient Texts: A Case Study about Ghewond's Armenian History

Marcella Tambuscio[1], Tara Lee Andrews[1,2]

[1]*Austrian Center for Digital Humanities and Cultural Heritage (ACDH-CH), Austrian Academy of Sciences, 1010 Vienna (Austria)*

[2]*University of Vienna, 1010 Vienna (Austria)*

Abstract

We present here a discussion about different methods to perform Named Entity Recognition tasks in order to extract geographic entities from the English translation of an Armenian text of the eighth century. Even though many tools are available and perform quite well with modern English, in this case they are only able to detect a very low percentage of the named geographic places. We compared four existing tools: NLTK and spaCy Python libraries, among the most used for NER tasks, TagMe, an entity linking tool that provide an annotation of found entities with Wikipedia pages, and Flair, a PyTorch library. We set these tools in order to select only geographical entities and we also tried two mixed methods: the best results on our data-set have been obtained by combining Flair and TagMe outputs with geographical clustering.

Keywords

named entity recognition, natural language processing, clustering, historical corpora

1. Introduction

As ancient and medieval texts increasingly become available in digital form, possibilities are opened for historians to perform not only traditional critical analysis, but also computationally-supported analysis of their contents. Alongside this, the explosion in technological possibilities clearly presents many opportunities to enrich the text, such as adding multimodal information or automatically extracting and highlighting some relevant information using natural language processing (NLP) techniques such as, for example, Named Entity Recognition (NER): identifying named entities that appear in unstructured texts and classifying them into categories (such as *person*, *location*, *organization* etc.). A related but slightly more complex task is Named Entity Linking (NEL), that adds the disambiguation step. We focus here on a particular sub-task of NER/NEL: the automatic extraction of geographical names in historical corpora. Although existing tools can achieve remarkable results with modern documents in the major world languages [32, 27], ancient and medieval texts pose particular challenges due to linguistic and geographical changes that take place over time, especially where the names or boundaries of those places or locations have changed. We provide here a case study using the English translation of an Armenian text of the VIII century, the *History* of Ghewond, for which (despite the robust support in general for English-language texts) existing NER toolkits performed quite badly. We tested four different tools (NLTK, Spacy, TagME, and Flair) to

CHR 2021: Computational Humanities Research Conference, November 17–19, 2021, Amsterdam, The Netherlands

marcella.tambuscio@oeaw.ac.at (M. Tambuscio); tara.andrews@univie.ac.at (T.L. Andrews)

0000-0003-2097-1333 (M. Tambuscio); 0000-0001-6930-3470 (T.L. Andrews)

extract geographical names and we compared the results. F-measures appeared to exhibit low values when confronted with the ones usually obtained with contemporary texts. We show that, by combining the outputs of the two best-performing tools, TagMe and Flair, with a geographical clustering, we can substantially improve their performance. We additionally modified the output of the tool TagMe that links the entities in the text to their respective Wikipedia pages, so that using the Wikipedia API we can select geographical places and directly extract the coordinates (when available) to create maps. Our main goal here is twofold: on the one hand, to compare the performance of four well-known NER toolkits on a challenging dataset (that is part of a larger collection of texts) and discuss the possible reasons of unsatisfying results; on the other hand, starting from the slightly more promising results of the toolkits trained on Wikipedia data, to explore new ways to retrieve information from this knowledge base to improve the results and lay the foundations to train a new and more performative model in the future.

2. Related Work

Named-entity recognition in historical texts is known to be a challenging problem [38] and especially for geographical places that have changed name over time or ceased to exist: empirical evidence suggests indeed that the more recent the texts, the more entities could be detected [13].

First, researchers have tackled the problem by developing different NER methods for identifying place references in a specific text or corpus, often enriched through the use of gazetteers. Most of them are rule-based or make use of the NLP Stanford NER tool: historical newspaper collections [12, 25, 26, 33, 28], literature [9, 10], British parliamentary proceedings [19], Turkish texts [26], Arabic historical texts [6, 36], Latin corpora [14], UK census data [30]. Similarly, some more general platforms have been created in order to be applied to different data-sets: the VARD tool [5] pre-processes historical corpora to propose modern equivalents alongside historical spelling variants; a digital geo-temporal gazetteer has been proposed in [29]; the Edinburgh Geoparser [2] and Recogito [37] recognise mentions of place names in text and assist in their disambiguation with respect to existing gazetteers. On the other hand, in [20] and [18] the authors provide a comparison of some existing geoparsers that make use of gazetteers, showing that they still have several limitations.

Secondly, in the last two decades researchers have been discussing the importance of Geographic Information Systems (GIS) technologies in the pursuit of historical research [16, 4, 17] and the necessity of introducing unsupervised methods which would allow a move from rule-based systems toward more data-driven approaches [11]. In [31] the authors discuss a combination of spatial analysis and natural language processing techniques in the field of archaeology. A mixed method that combines NLP and geospatial clustering has been proposed in [23] to identify places in housing advertisements: even if the inputs here are not historical data, the challenge is similar since many local place names either have not been registered in gazetteers or appear in abbreviated forms that do not appear.

Thirdly, some recent work has suggested that results improve when several tools are combined. In [39] the authors propose a method that considers five NER tools through a voting system, which produced a better performance than any single tool. Similarly, GeoTxt [24] is a geoparser for unstructured streaming text that supports multiple NER methods.

3. The Dataset

The *History* of Ghewond covers events in and around Armenia from ca. 632 to 788, focusing on the Arab domination of the region and especially the transition from Umayyad to Abbasid rule and its effects on Armenian politics and society. It is a short text (25K words) but the geographical range is broad, covering the centers of power of the Caliphate, the territories both of the former Roman Armenia and the so-called Persarmenia, as well as references to Byzantine and Khazar places. The impetus for the study was the desire to identify and collect the context for place names in works of Armenian history, while being fully aware that NER tools for the Classical Armenian language are not in a particularly advanced state of development. The solution that seemed obvious was to analyse modern English translations of the texts. For the initial attempt we used the English translation of Ghewond's *History* made in 2006 by Robert Bedrosian, who is well known for his translations of several works of Armenian history. Bedrosian's translation style is to stay as close as possible to the original text phrasings, and in particular to render all proper names in a direct transliteration from their forms in the Armenian alphabet. While this is a welcome and helpful translation strategy from the perspective of historians of the medieval Caucasus, the place names themselves can be difficult both for untrained human readers and for neural networks to recognise. Adding to the complication is the fact that, in medieval Armenian society, territories and their ruling clans often carried the same names; this means that, for example, any occurrence of a name such as "Rshtunik" must be examined for context to determine whether this is a mention of a place or a group of persons. In order to compare the outputs of several NER approaches, we manually created a gold standard to list the geographical entities and their occurrences: we found 199 entities with 303 total occurrences in the text. The most frequent are *Armenia, Judaea, Vaspurakan, Byzantine territory, Damascus, Byzantium, Asorestan.*

4. Methods

All the codes and the data can be found on GitHub [1].

4.1. NER tools

First of all we briefly describe here the tools that we used.

NLTK (Natural Language Toolkit)[2] and **spaCy**[3] are libraries for NLP written in Python. They were originally developed for English and perform tasks such as tokenization, classification and part-of-speech tagging. NLTK [7, 8], developed by the University of Pennsylvania, is intended to support research while spaCy [22], published under the MIT license, is more application-oriented. The set of labels offered by the standard English models of NLTK and spaCy libraries include a `GPE` label for geopolitical entities as countries, cities, states and a `LOC` label for physical locations as mountain ranges, rivers, seas. We selected entities detected in our text with both labels.

Flair is a relatively new library, open source and developed in Python by the Humboldt University of Berlin and Zalando Research, that offers support for common NLP tasks including Named Entity Recognition [1]. This service seems to be more powerful than spaCy, but it must

[1] https://github.com/tambu85/ancient_text_NER
[2] https://www.nltk.org/
[3] https://spacy.io/

be observed that Flair is (a bit) slower and is currently available for only a few languages. For our purposes we selected only entities classified with the `LOC` label.

TagMe[4] is an impressive entity linking tool, developed by the University of Pisa[15], that identifies meaningful *spots* in an unstructured text and links each of them to a pertinent Wikipedia page. For this reason it has been used for disambiguation tasks [35]. TagMe usually performs very well with short texts, but it can also be used on longer ones. Given an input text, the TagMe API [5] provides a list of annotations, meaning a list of pairs *(spot,entity)*, where each *spot* is a substring of the input text and each *entity* is a reference to a unique Wikipedia page representing the meaning of that spot, in that context. TagMe computes for each entity a *link probability lp* that measures how frequently the spot text is used to link exactly that entity page. Moreover, TagMe associates a value ρ (rho) to each annotation, which estimates a confidence score of the annotation among the possible entities. In TagMe there is also a parameter that can be used to fine-tune the disambiguation process, either to select the most common topics for a spot or to take the context of each spot more into account (we selected this second option). This parameter could be useful when annotating particularly fragmented text, such as tweets, where it would be better to favor the most common topics because the context is less reliable for disambiguation. Supported values are floats in the range [0,0.5], default is 0.3. It should be noted that TagMe itself does not provide any sort of classification of the entities and it was not designed for this task. Nevertheless, we noticed that it was able to detect many geographical entities that the other tools missed: one reason could be that Wikipedia often reports also the ancient names of places. Then we added our own basic classification by filtering the results using a SPARQL query through the Wikidata Query Service[6] to select only geographical places. Moreover, when the geographical coordinates were available in the page, we added them to the output: in the appendix we provide all the statistics about this extension of TagMe.

4.2. Mixed approaches

We applied the above-mentioned tools to our historical text and evaluated the results, comparing them with a gold standard of manual annotations: in the next section we will provide the well-known evaluation measures. None of the tools provided exceptional results, but the best ones were obtained by TagMe and Flair, even if these tools are significantly slower than spaCy and NLTK. We then tried to improve the results, proposing two other methods:

- **M1** we simply considered the union of the results of TagMe and Flair;
- **M2** we ran a geographical clustering to discard non-meaningful TagMe results and then we considered the union of these filtered results with Flair.

Note that in M2 we are considering only the entities with coordinates, and among them we perform another selection with clustering. We chose DBSCAN, a popular density-based clustering algorithm, and we set the parameters ($\epsilon = 10$, $minpts = 30$) following the standard procedure and computing the k-nearest neighbors (k-NN) for different values of k [21]. In section 5 we show how this is a valid approach to remove many false positives from the TagMe output.

[4] https://sobigdata.d4science.org/web/tagme/tagme-help
[5] An official wrapper for Python is available here: https://github.com/marcocor/tagme-python
[6] https://query.wikidata.org/

4.3. Validation and Evaluation

We selected the LOC and GPE entities detected by NLTK, spaCy and Flair and the Wikipedia entities labeled as geographical ones by our SPARQL query. Then we validated the results by comparing them with a gold standard that we produced manually. We had to define some rules for some corner cases:

- *Ethnonym references.* Many entities are of this form: `Armenian (lords)`, `land of the Aghuanians` or `Byzantine territory`. Nevertheless, we have to report that most of the time such expressions are not detected by any of the methods described (with some exception recognized by TagMe and Flair). We decided to consider as true positives only expressions that include a geographical element such as `the country of Armenians` or `Byzantine territory`.
- *Temporally ambiguous results.* An example is `Syria`, for which TagMe occasionally provides a link to the modern republic, while in the text the author is referring to the Roman province; these two geographical areas overlap but are not coterminous. In the Appendix we report the percentage of the completely matching entities. In this context, in order to compare TagMe usefully with the other tools, we did not consider the entity linking (EL) evaluation, but only the NER task: the annotation is labeled as correct if the spot is indeed a location in essentially the right place. Conversely, when we had to deal with geographical clustering (mixed method M2) only the entities with the correct coordinates were considered valid. Another example is the entity `Iberia` that TagMe associates (in different occurrences) to two different Wikipedia pages:`Iberian Peninsula` and `Kingdom of Iberia`. For NER task, both are considered correct since `Iberia` is a geographical entity in the text, while for EL task in M2 only the second is considered correct.

To evaluate the results we used the three common measures in classification tasks: precision, recall and F1-measure [3, 34]. Here we briefly review their formula and meaning in our context. Our task is to find geographical entities and we have a gold standard (the correct list) to compare them. For each tool we can count:

- TP (true positives), number of entities correctly labeled as locations;
- FP (false positives), number of entities incorrectly labeled as locations;
- FN (false negatives), number of entities which were not labeled as locations but should have been.

From this we can compute precision and recall:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

These measures are not particularly useful metrics if used individually, since a high value for one of them does not necessarily represent a good performance. Instead, we consider the F1-measure (or F-score), the harmonic mean of the previous measures, to evenly weight them:

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

The F1-measure produces values in $[0, 1]$ where 1 represents perfect precision and recall.

Tool	Precision	Recall	F1-measure
NLTK	0.404	0.405	0.404
spaCy	0.668	0.385	0.488
TagMe	0.537	0.473	0.503
Flair	**0.772**	0.677	0.721
M1	0.589	**0.796**	0.677
M2	0.738	0.760	**0.748**

Table 1
Precision, recall, and F-measure for each method tested. M2, a combined approach of Flair, TagMe and geographical clustering gives the best F1-score.

5. Results

In Table 1 we give the evaluation measures that were computed for the different methods that we used to detect entities defining locations in Ghewond's *History*.

The values are surprisingly low if compared with the average results on known datasets, where these NER tools usually give F-measures that far exceed 0.9. These low values represent a clear example of how the NER task can still present research challenges, especially on historical texts. Some comments about the comparison:

- NLTK is the worst-performing tool, with very low values for each measure;
- spaCy performs a little better than NLTK in terms of precision, meaning that there is a smaller fraction of FP (entities incorrectly labeled as locations) but the recall is very low due to a high number of missed entities (FN);
- TagMe performs slightly better than the previous two, with similar values for precision and recall;
- Flair is the best among the four tools and exhibits the best precision score among all the proposed methods.

It is important to note that low recall scores are partially a consequence of our choice to consider as TP expressions like `country / land of the Armenians` since they appear many times in the text but are only rarely captured by the tools. This is not necessarily a problem, since we could manually tune many of these tools by adding some specific context-based pattern matching rules to detect such expressions.

While Flair clearly exhibited the best performance, it must be observed that TagMe detected many entities that were not captured by the other three, particularly locations of places referred to using obsolete names (for instance the Roman Province `Judaea` or the medieval name of Istanbul, `Constantinople`). This is due to the fact that TagMe links entities in the text to their Wikipedia entries, which are often reachable under the several names by which these places were known during different eras. Moreover, by adding the geographical coordinates we were able to enrich the output and, since Wikipedia is constantly growing, when repeating the experiment on Ghewond's *History* or other similar texts we can expect to obtain ever better results.

Since there was a significant subset of entities detected only by TagMe, our next experiment was to use a mixed approach M1, where we considered the union of the outputs given by Flair

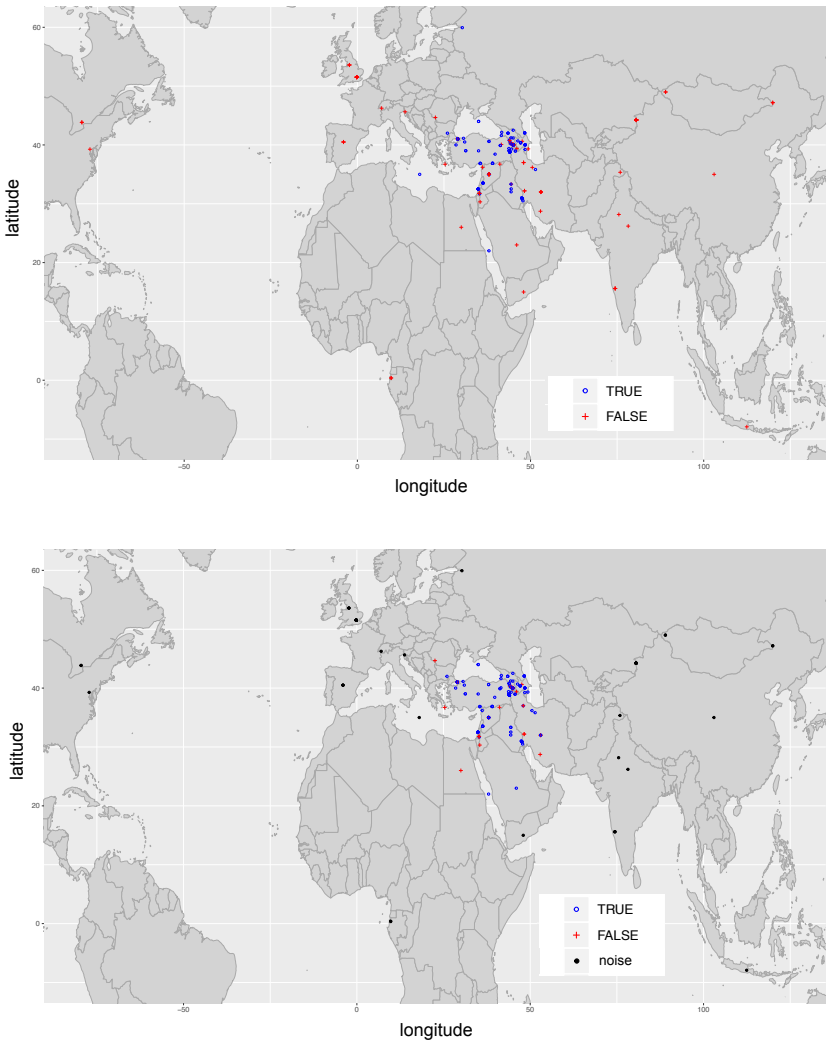

Figure 1: Comparison of the set of entities with coordinates found by TagMe (top) and result of DBSCAN algorithm ($\epsilon = 10$, $minpts = 30$) over the same set (bottom). In particular we want to show that discarding the noise discriminated by the clustering algorithm we can remove a significant portion of FALSE POSITIVE (red) and isolate the TRUE POSITIVE (blue).

and TagMe. This gave the best recall score but a lower precision with respect to Flair by itself: this means that (as would be expected) Flair and TagMe together are less likely to miss locations, but still produce a large number of entities incorrectly labeled as places. Since Flair had a high value for precision, the imprecision of the M1 approach clearly originates in TagMe. To partially mitigate this problem, we made use of the coordinates returned by TagMe: we selected only those returned entities with geographical coordinates and we ran a clustering algorithm to detect noise.

In Figure 1 (top) we plot the validated entities with coordinates that were found within the text using TagMe and Wikipedia queries: blue circles are the true positives, while red crosses are the false positives. The validation used in this case is a strict one: if an entity appears as a location, but the coordinates are wrong, we label it as a false positive. We can observe that the true positives are quite clustered so we ran DBSCAN, an unsupervised clustering algorithm:

we remark that (as yet) we are not interested in how many or which clusters the algorithm detected; we have merely used the clustering to remove many false positives, labeled as noise. In Figure 1 (bottom) we show that this method works quite well, which led us to consider only the entities within the clusters for the M2 method.

Finally we tried the second mixed approach M2, in which we consider (as in M1) the union of the outputs given by Flair and TagMe with a clustering step. As shown in the last row of Table 1, M2 was the best among the six approaches: even if it is not optimal, we obtained the highest F-score and a good balance among precision and recall. In the appendix we discuss the results in detail, considering the advantages and disadvantages of the various approaches.

6. Conclusions

In this paper we propose a case study on automatic detection of geographical entities in a corpus originally written in a medieval minority language. Although an English translation of the text was available for use, the fact that the place names in the text refer to a totally different geopolitical system is actually an obstacle for a system trained on modern English, meaning that "out-of-the-box" NER tools fail most of the time. Indeed, NLTK and spaCy, the most well-known tools for NER tasks, obtained very low F-measures on our corpus. Our attempts with TagMe (designed for Entity Linking) and Flair, two different tools both trained on Wikipedia data, do provide better results. Although these two tools are significantly slower than NLTK and spaCy, and the execution time can be very important for NLP tasks on streaming and real-time data, we do not consider this a major problem for a NER task run on a historical text, since it is likely to be ran only once. Moreover, the modified version of TagMe that we have devised is even slower due to the use of the Wikipedia API to get coordinates, which is also impacted by a rate limitation on requests. Concerning the quality of performance of these tools, we must bear in mind that, since Wikipedia is an online encyclopedia maintained by a community of volunteer editors, it continuously changes over time: this means that the results of a repeated analysis could vary (hopefully for the better), or even that larger common knowledge databases could offer alternative solutions in the future. Finally, we tried two mixed approaches and found that by combining Flair and TagMe results with clustering techniques we were able to significantly improve their performance. It is, however, important to note that this approach can also depend on the type of data: since we knew that most of the events described in the text happened in a circumscribed area, clustering was helpful in that it allowed us to discard some entities that were wrongly classified as places. This could also be the case for other historical texts, but more detailed research on larger datasets could provide new insights about the usefulness of geographical clustering for entities.

We see different future steps for this research line: (i) this is an initial a case study, so more tests are needed on other corpora which could also include some comparison with other tools that gave similar F1 scores on other datasets [24]; (ii) the detection process using TagMe and geographical clustering could still be improved; (iii) the mixed approach of Flair and TagMe (maybe perhaps with additional metadata from Wikipedia) could also be used for other types of entities such as person and organisation names.

Acknowledgments

Thanks to the developers of NLTK, spaCy, Flair and TagMe.

References

[1] A. Akbik, D. Blythe, and R. Vollgraf. "Contextual String Embeddings for Sequence Labeling". In: *COLING 2018, 27th International Conference on Computational Linguistics*. 2018, pp. 1638–1649.

[2] B. Alex, K. Byrne, C. Grover, and R. Tobin. "Adapting the Edinburgh geoparser for historical georeferencing". In: *International Journal of Humanities and Arts Computing* 9.1 (2015), pp. 15–35.

[3] R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*. Vol. 463. ACM press New York, 1999.

[4] T. J. Bailey and J. B. Schick. "Historical GIS: enabling the collision of history and geography". In: *Social Science Computer Review* 27.3 (2009), pp. 291–296.

[5] A. Baron and P. Rayson. "VARD2: A tool for dealing with spelling variation in historical corpora". In: *Postgraduate conference in corpus linguistics*. 2008.

[6] M. A. Bidhendi, B. Minaei-Bidgoli, and H. Jouzi. "Extracting person names from ancient Islamic Arabic texts". In: *Proceedings of Language Resources and Evaluation for Religious Texts (LRE-Rel) Workshop Programme, Eight International Conference on Language Resources and Evaluation (LREC 2012)*. 2012, pp. 1–6.

[7] S. Bird. "NLTK: the natural language toolkit". In: *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*. 2006, pp. 69–72.

[8] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. 1st. O'Reilly Media, Inc., 2009.

[9] L. Borin, D. Kokkinakis, and L.-J. Olsson. "Naming the past: Named entity and animacy recognition in 19th century Swedish literature". In: *Proceedings of the Workshop on Language Technology for Cultural Heritage Data (LaTeCH 2007)*. 2007, pp. 1–8.

[10] J. Brooke, A. Hammond, and G. Hirst. "GutenTag: an NLP-driven tool for digital humanities research in the Project Gutenberg corpus". In: *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*. 2015, pp. 42–47.

[11] T. Brown, J. Baldridge, M. Esteva, and W. Xu. "The substantial words are in the ground and sea: computationally linking text and geography". In: *Texas Studies in Literature and Language* 54.3 (2012), pp. 324–339.

[12] G. Crane and A. Jones. "The challenge of virginia banks: an evaluation of named entity analysis in a 19th-century newspaper collection". In: *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*. 2006, pp. 31–40.

[13] M. Ehrmann, G. Colavizza, Y. Rochat, and F. Kaplan. "Diachronic evaluation of NER systems on old newspapers". In: *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*. Conf. Bochumer Linguistische Arbeitsberichte. 2016, pp. 97–107.

[14] A. Erdmann, C. Brown, B. D. Joseph, M. Janse, P. Ajaka, M. Elsner, and M.-C. de Marneffe. "Challenges and solutions for Latin named entity recognition". In: *COLING 2016: 26th International Conference on Computational Linguistics*. Association for Computational Linguistics. 2016, pp. 85–93.

[15] P. Ferragina and U. Scaiella. “Tagme: on-the-fly annotation of short text fragments (by wikipedia entities)”. In: *Proceedings of the 19th ACM international conference on Information and knowledge management*. 2010, pp. 1625–1628.

[16] M. F. Goodchild and L. L. Hill. “Introduction to digital gazetteer research”. In: *International Journal of Geographical Information Science* 22.10 (2008), pp. 1039–1044.

[17] I. N. Gregory and A. Hardie. “Visual GISting: bringing together corpus linguistics and Geographical Information Systems”. In: *Literary and linguistic computing* 26.3 (2011), pp. 297–314.

[18] M. Gritta, M. T. Pilehvar, N. Limsopatham, and N. Collier. “What’s missing in geographical parsing?” In: *Language Resources and Evaluation* 52.2 (2018), pp. 603–623.

[19] C. Grover, S. Givon, R. Tobin, and J. Ball. “Named Entity Recognition for Digitised Historical Texts.” In: *Lrec*. 2008.

[20] C. Grover, R. Tobin, K. Byrne, M. Woollard, J. Reid, S. Dunn, and J. Ball. “Use of the Edinburgh geoparser for georeferencing digitized historical collections”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 368.1925 (2010), pp. 3875–3889.

[21] M. Hahsler, M. Piekenbrock, and D. Doran. “dbscan: Fast density-based clustering with R”. In: *Journal of Statistical Software* 91.1 (2019), pp. 1–30.

[22] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd. *spaCy: Industrial-strength Natural Language Processing in Python*. 2020. DOI: 10.5281/zenodo.1212303.

[23] Y. Hu, H. Mao, and G. McKenzie. “A natural language processing and geospatial clustering framework for harvesting local place names from geotagged housing advertisements”. In: *International Journal of Geographical Information Science* 33.4 (2019), pp. 714–738.

[24] M. Karimzadeh, S. Pezanowski, A. M. MacEachren, and J. O. Wallgrün. “GeoTxt: A scalable geoparsing system for unstructured text geolocation”. In: *Transactions in GIS* 23.1 (2019), pp. 118–136.

[25] K. Kettunen, E. Mäkelä, T. Ruokolainen, J. Kuokkala, and L. Löfberg. “Old content and modern tools-searching named entities in a Finnish OCRed historical newspaper collection 1771-1910”. In: *arXiv preprint arXiv:1611.02839* (2016).

[26] D. Küçük et al. “Named entity recognition experiments on Turkish texts”. In: *International Conference on Flexible Query Answering Systems*. Springer. 2009, pp. 524–535.

[27] J. Li, A. Sun, J. Han, and C. Li. “A survey on deep learning for named entity recognition”. In: *IEEE Transactions on Knowledge and Data Engineering* (2020).

[28] S. Mac Kim and S. Cassidy. “Finding names in trove: named entity recognition for Australian historical newspapers”. In: *Proceedings of the Australasian Language Technology Association Workshop 2015*. 2015, pp. 57–65.

[29] H. Manguinhas, B. Martins, and J. Borbinha. “A geo-temporal web gazetteer integrating data from multiple sources”. In: *2008 Third international conference on digital information management*. Ieee. 2008, pp. 146–153.

[30] P. Murrieta-Flores, A. Baron, I. Gregory, A. Hardie, and P. Rayson. “Automatically analyzing large texts in a GIS environment: The registrar general’s reports and cholera in the 19th century”. In: *Transactions in GIS* 19.2 (2015), pp. 296–320.

[31] P. Murrieta-Flores and I. Gregory. “Further frontiers in GIS: Extending spatial analysis to textual sources in archaeology”. In: *Open Archaeology* 1.open-issue (2015).

[32] D. Nadeau and S. Sekine. “A survey of named entity recognition and classification”. In: *Lingvisticae Investigationes* 30.1 (2007), pp. 3–26.

[33] C. Neudecker, L. Wilms, W. J. Faber, and T. van Veen. “Large-scale refinement of digital historic newspapers with named entity recognition”. In: *Proc IFLA Newspapers/GENLOC Pre-Conference Satellite Meeting*. 2014.

[34] D. Powers. “Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation”. In: *Journal of Machine Learning Technologies* 2.1 (2011), pp. 37–63.

[35] M. Rovera, F. Nanni, S. P. Ponzetto, and A. Goy. “Domain-specific named entity disambiguation in historical memoirs”. In: *CEUR Workshop Proceedings*. Vol. 2006. Rwth. 2017, Paper–20.

[36] K. Shaalan. “A survey of arabic named entity recognition and classification”. In: *Computational Linguistics* 40.2 (2014), pp. 469–510.

[37] R. Simon, E. Barker, L. Isaksen, and P. de Soto Cañamares. “Linking early geospatial documents, one place at a time: annotation of geographic documents with Recogito”. In: *e-Perimetron* 10.2 (2015), pp. 49–59.

[38] S. Van Hooland, M. De Wilde, R. Verborgh, T. Steiner, and R. Van de Walle. “Exploring entity recognition and disambiguation for cultural heritage collections”. In: *Digital Scholarship in the Humanities* 30.2 (2015), pp. 262–279.

[39] M. Won, P. Murrieta-Flores, and B. Martins. “ensemble named entity recognition (ner): evaluating ner Tools in the identification of Place names in historical corpora”. In: *Frontiers in Digital Humanities* 5 (2018), p. 2.

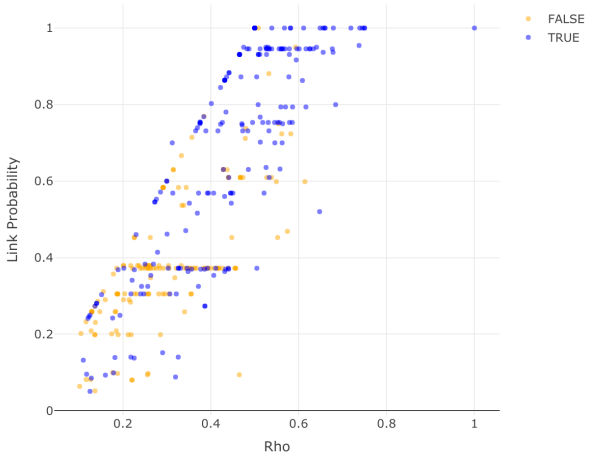

Figure 2: Scatterplot of the detected entities by TagMe according to their values of confidence and link probability. Each point (ρ, lp) represents a detected entity with confidence ρ and link probability lp; its color discriminates true positive (blue) and false positive (false).

A. Detailed analysis of TagMe results

In the course of our work we also explored whether we can use the values of the parameters ρ and lp to better tune the TagMe tool. These values can be used to discard annotations that are below a given threshold. First of all we visualize the True Positive (TP) and False Positive (FP) in a scatter plot (see Fig. 2). Even if there are not well-separated clusters, we can see that there is a higher density of FP for low values of ρ and lp.

We then computed how precision, recall and F1-measures vary when moving the thresholds of ρ and lp in $[0, 1]$. When we fix a threshold τ all the TP obtained for values lower than τ become FN (missed entities). Results are shown in Fig.3 (varying one parameter at a time) and Fig.4 (varying both parameters). As we can see, the F-measure slowly decreases at the beginning and then falls. This accords with the recommendation of the TagMe authors, who indicate values between 0.1 and 0.3 as reasonable threshold for ρ. In our case, we set 0.1 as threshold for ρ and 0.05 for lp (the recommended standard), but such an approach should be repeated in other data sets to explore the role of these parameters.

Finally in Tab.2 we report the rates of TP associated with the correct Wikipedia entity and coordinates, obtained with a double manual validation: TagMe identified the right entity 81% of the times over all detected geographical entities and provided coordinates for 71% of them. By combining both we obtained the result that 59% of the detected entities are linked to the right Wikipedia pages that exhibit coordinates. This could surely help in automatically providing a map of the different entities.

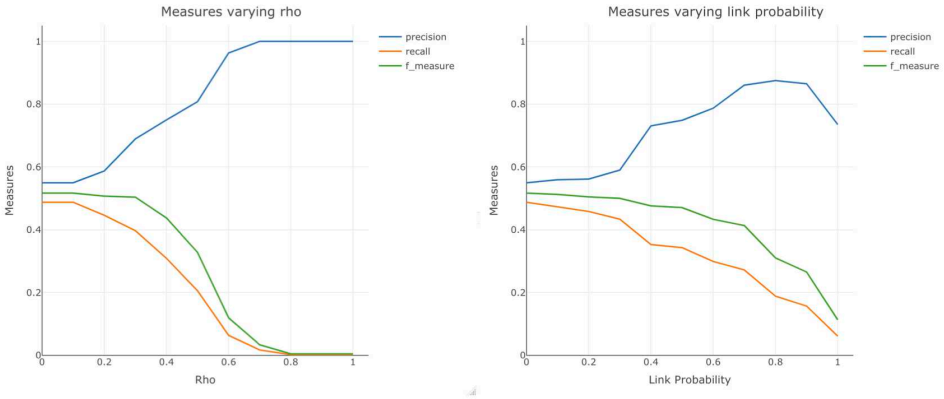

Figure 3: Precision, Recall and F1-measure changes fixing a threshold on ρ (left) and on lp (right) to discard some results.

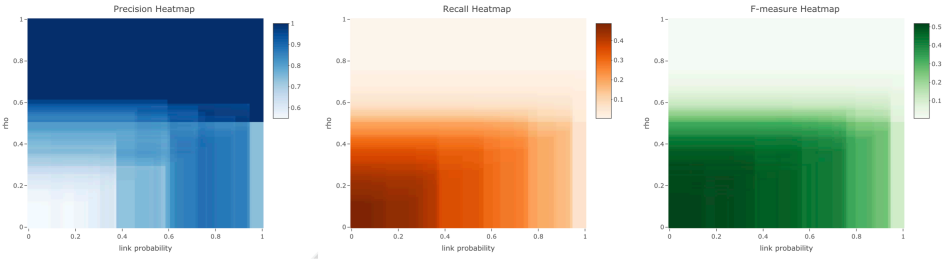

Figure 4: Heatmaps that show how the Precision, Recall and F1-measures change by fixing a threshold on both ρ and lp (right) to discard some results. For each point (ρ_i,lp_j) we compute the measures selecting only values that exhibit a confidence $\rho \geq \rho_i$ and a linking probability $lp \geq lp_j$.

Table 2
Rates of entities correctly detected in Wikipedia, entities with coordinates and entities correctly detected in Wikipedia with coordinates.

Right Entity Linking	Coordinates	Right Entity Linking and Coordinates
0.81	0.72	0.61