DHQ: Digital Humanities Quarterly
2021
Volume 15 Number 1

Inferring standard name form, gender and nobility from historical texts using stable model semantics

Davor Lauc <davor_dot_lauc_at_ffzg_dot_hr>, Chair of Logic, Department of Philosophy, Faculty of Humanities and Social Sciences, University of Zagreb
Darko Vitek <dvitek_at_hrstud_dot_hr>, Department of History, Centre of Croatian Studies, University of Zagreb

Abstract

In this paper, we attack the problem of parsing name expressions and inferring standard name form, gender and nobility status from serial historical sources. This is a small but important part of modelling historians' analysis of such sources, as they extract a lot of information from the names in text, and this information constrain their search. The task of parsing proper names seems to be easy, but it is a hard problem even for the modern languages, and even more challenging for the languages of historical sources. The test case used for the research was from the middle 19th century census for the old town centre of Zagreb. In order to evaluate and compare the fitness of the probabilistic and rule-based models for the task of inferring standard name form, both conditional random field (CRF) and rule-based models based on stable model semantics (Answer Set Programming Rules) were developed. Our results indicated that the rule-based approach is more suitable for inferring standard name forms from historical texts than the more widespread statistical approach.

Introduction

The necessary condition of many endeavours in the digital humanities domain is the preparation of data in a suitable form. In the case of historical demography, this means extracting structured data about people from sources like national censuses, cadastral data, tax lists, etc. These sources are unstructured information, concealed formats that are hard to decode and laden with ambiguity. This task is often performed manually by a trained historiographical researcher, who scrutinizes archived documents. It is an extremely tedious exercise that is prone to error. 1

Due to the availability of many historiographical sources in various digital formats – from scans to transcripts – it is possible that a computation model able to achieve this task could be developed. Ideally, this system would be able to transform unstructured data, like scans of historical documents, into structured formats. In this sense, it would be great to have historical data in highly structured form, with ambiguity reduced on every level – the semantic web is a good representational format to do this.[1] 2

However, the implementation of such models comes with many challenges, even when transcripts of sources are available in digital or printed formats, and the problematic recognition of indecipherable handwritten text [Vamvakas 2008] can be omitted. 3

One of the first, seemingly trivial, challenges relates to parsing name expressions that include individuals' names, titles and occupations, and inferring basic facts about persons, such as gender and some aspects of their social statuses. This is often a prerequisite for more advanced processing, for example named entity resolution (record linkage), ontology construction, etc. Even when more contemporary data sources are involved, the ambiguity, multitude and various combinations of first name/last name/titles that are in use can make this task quite difficult to model. Sometimes, there is just not enough information available in text to reach reliable conclusions, and only an educated guess is possible. In the case of historical source transcripts, the task is even more challenging because many of the names and 4

personal titles involved are now extinct and cannot be found in modern dictionaries. Furthermore, there are no standard transcriptions of names, and those transcripts that exist are often mottled and dirty.

Not much research on this particular topic has been done within the digital humanities community, but the authors expect that this will become an active field of research. As Anna Foka claims: 5

> The imminent assessment and representation of historical data has admittedly challenged the boundaries of historical knowledge and generated new research questions. [Foka 2018]

The related problem of entity resolution, and its importance to digital humanities for have been has been researched more extensively [Johannessen at al 2005] [Boren 2007] [Heckmann at all 2014] [De Wilde 2017]. Proper solution of this particular problem of structuring proper names can have important application to many digital humanities endeavours, from improvements of handwriting recognition systems where correct parse of a proper name can be used to improve the loss function, to usage in multilingual family narrative generation from genealogical data [Lauc and Vitek 2018].

Materials

The primary type of the historical sources in the scope of this research are the so-called serial sources. These are sources like parish books (one of the biggest serial sources in European history), tax lists, and censuses. One of their important characteristics is systematic repetition of structure, which makes them a kind of predecessors of modern database systems, but unfortunately not in a modern structured format. Typical historiographical method of processing such sources includes a taxing process of translating them into structured forms, and it often happens that even when this is done, data are underused due to number of reasons, such as low-tech solution, large number of errors, etc. It often takes a lifetime of one lonely researcher to finish analysing just one such source, for example one tax census [Vrbanus 2010]. 6

Characteristics of Serial Sources in Croatian Demography

During the Middle Ages, Croatian state had undeveloped state institutions, which was mostly caused by weak central government. This was the main reason why Ottoman Empire successfully conquered it in the 15th and 16th century. Therefore, the number of sources from that period is quite limited and they are almost exclusively in Latin.[2] 7

Test Case

The test case used for the research was from the middle 19th century census for the old town centre of Zagreb. This census, which was performed in 1857, was the first modern census in Croatia. It is a valuable source of information about the social structure of the 19th century Zagreb. A systematic analysis of the data has yet to be performed due to a number of difficulties. At the time of the census, Croatia was a part of the Habsburg monarchy and the language of the census is German. As the Latin language was dominant in the public service, many Latin name forms were Germanized, and many Czech, Slovak, Italian and German names were Croatized or Latinized. Additionally, there was no standard order of first and last names, many of the first and last names were not separated, and various additional notes were not clearly separated from names, especially when it comes to members of the nobility. Because of all these difficulties, the manual transformation of this source into structured data represents a very tedious and error-prone task. 8

Our goal was to develop a highly accurate model that can parse name expressions and infer standard names from Zagreb's middle 19th century census and similar historical texts. There was also a requirement to predict gender and nobility of the person from the name expression. The transcription was available as a text file, and after some standard pre-processing and chunking, 1755 records were extracted, including expressions of person names, dates of births, occupation and similar. We decided to apply standard natural language processing tools to this text. 9

Methods and Methodology

Our hypothesis was that the best model for this task could be achieved by combining a probabilistic approach with a 10

rule-based approach in the framework of Answer Set Programming.

Applying Standard NLP Pipeline in a Historical Text

A common approach to tackle the problem of transforming unstructured images to structured information is designing a natural language programming (NLP) pipeline, wherein the first step includes optical character recognition (OCR)/handwriting recognition (HWR) or transcription. Unfortunately, both of these processes are laden with challenges, and more than often they include a lot of decrypting. The result of converting a historical source to text is usually very messy, and it is commonly referred to as a “dirty” text. As further processing requires relatively clean text, the dirty text usually needs to be corrected – this includes spell-checking, joining separated word parts, etc. 11

The next phase is tokenisation, i.e. splitting text into tokens, usually words. However, the situation is often quite complicated. Defining what exactly counts as a token (e.g. does it include full stops) is an important decision that can have big impact on further steps. Ready-available tokenizers (usually rule-based) did not perform well on our test case. Therefore, we have applied an aggressive tokenisation in our research, splitting everything into sequences of letters vs. non-letters. Another connected step in the NLP pipeline is the segmentation of text, usually into sentences. Although this step is not exactly applicable to our test set, it is a well-known fact that sentence splitters do not perform well on historical texts, as they are trained on the modern ones [Petran 2012]. 12

The following step would be name-entity recognition (NER), i.e. marking the beginning and end of named entities and classifying each such expression as person, organisation, place, temporal expression, etc. Since serial sources consist mostly of such named entities, successful NER would chunk almost the entire text. Unfortunately, modern NER systems do not perform well on historical text. We have tested some of the best NER systems like Stanford NER, which has an f-score of over 90% on modern texts, but it has performed badly on our test case. We have not the calculated exact measure, but the results were so flawed that it seemed needless to do a formal evaluation. 13

Another important standard task in this context would be Entity resolution, also called record linkage or record deduplication, where different occurrences of the name of same entities are connected. For example, in parish church books, one person might first be baptised, then married, then mentioned as godfather and finally be deceased. It is worth identifying all these names as referencing the same object, in this case the same person. 14

Finally, the last phase would be relation or relationship extraction, i.e. extracting relations among entities in the text, for example if a person is born at some place, married to someone, etc. In the broader sense, relation extraction can also include inferring the relationships. For example, if we have a relation being *father of* between a father and two daughters, we can infer relation *being sister* between them. Similarly, as the relation *being mother of* is functional, when it occurs between one entity and two different entities, the system can identify those entities as the same. This is something that is standardly done by semantic web technologies. 15

Naturally, since the NER systems did not perform well on the test case, it was impossible to apply the existing relationship extraction systems. 16

It is worth mentioning than an alternative approach to a segmented NLP pipeline could be a technique called joint inference [Poon and Domingos 2007]. In joint inference all levels of processing are performed in the same time, with constraints and information on all levels used for inferring the most probable inference. This approach is more similar to the process of analysis performed by a human researcher. 17

Proper Names Parsing

The segmented parts of the census were available, so our research started with analysing what would a historian working with such data do with them. We first wanted to understand how this process is performed by a human, in order to build an effective computational model. The first obstacle we encountered was understanding proper names. When historians analyse such sources, they often unconsciously extract a lot of information from the name, and these information constrain their search by reducing ambiguity. For example, if two persons in the same part of the text have 18

the same surname, they are probably related; if someone has a female name, it is very improbable that that person is a godfather, etc. Therefore, it is worth extracting these information from the name, especially in the context of joint inference, where all available information should be used to resolve ambiguity.

Parsing of proper names, such as analysing inner structure of names, is not a standard part of NLP pipeline. Parts of text containing proper names are usually recognised by a NER subsystem, and other techniques are used for entity resolution and relation extraction subsystems. The task of parsing proper names does seem to be easy; in the case of personal names, one just has to use regular expressions – first token is first name and second token is last name. After that, the only thing left to do is to check the ending of the first name to classify the person as a female or a male. In reality, situation is much more complex. Even when only the modern names are analysed, there is no easy solution to the name parsing, at least no ready-available system. 19

To illustrate the problem, even in modern languages there is a great variety of names, name forms and cultural conventions. In the case of famous Icelandic singer Björk Guðmundsdóttir, her second name is not really a surname and one should address her with full name. This means that you would find her listed in address book under letter B not G. Similar case is with Arabic, Chinese, Russian, Polish, Serbian and other names that usually start with the last name. Spanish people traditionally have four names, and are addressed by the third one, and the situation in even more complicated in Brazilian Spanish. 20

The situation with names is even more complex in historical contexts. Until the second half of the 18th century, which is relatively recent, there were no standard name forms in many European countries. In the example of the first Croatian king, Kralj Tomislav, “Kralj” means king – it is a title, not a first name, and “Tomislav” is the first name. In the modern language, however, “Kralj” is a quite common last name. The situation gets even more complicated with noble titles, maiden names, etc. 21

It would be very helpful to have an accurate proper name parser, the results of which could be used in a more advanced analysis. To the best of the authors’ knowledge, this problem did not receive much attention in NLP related communities. 22

Proper Names Parsing State of the Art

Parsing name records into constituent parts can be modelled as a sequence labelling problem, viewed as a special case of part-of-speech tagging and shallow parsing. Although this particular problem has not received much attention in recent literature, extensive work has been done on the related and more general problems of Part-Of-Speech (POS) tagging, shallow parsing and named entity recognition [Graves 2012] [Osborne 2000] [Ndeau and Sekine 2007]. 23

Early sequence labelling systems were rule-based; for example, those developed by Brill [Brill 1992], which are still used in some application domains [Chiticariu 2010]. Today, the best performing models are probabilistic, and are generally based on probabilistic graphical models. In particular, models that use conditional random fields represented the state-of-the-art models [Sha and Pereira 2003] [Viet Cuong at al 2014], until the recent usage of deep learning models [Devlin at al 2018]. 24

However, studies on the application of probabilistic models on historical texts have yet to yield satisfactory results. It is very tedious to annotate historical texts, especially when many different sources have to be analysed and the reuse of existing training datasets is not a feasible option. Another reason why there is a need to consider alternatives to the probabilistic approach is that, due to the noisiness of historical sources, the integration of sequence labelling with a joint inference model [Sha and Pereira 2003] is promising as an alternative to the use of a traditional language processing pipeline. Joint inference can reduce OCR ambiguities, and an approach that combines text correction and sequence labelling with the higher-level syntax, semantics and historiographical constraints is more representative of the way in which a human historiographer would perform the task. For example, the probability of social status depends on location, members of the household and place of origin, and ambiguous last names can be resolved by family member records. Although joint inference is compatible with the probabilistic approach [McCallum 2009], the rule-based approach seems more promising in this domain. As the Markov Logic Networks, a framework that Sha and Pereira [Sha 25

and Pereira 2003] used, does not readily scale to this kind of problem, we selected a rule-based framework that was based on stable model semantics.

Answer Set Programming

Stable model semantics can be viewed as the semantics of logic programming. So the rules can use (almost) the full expressivity power of the first-order logic. A further benefit relates to the non-monotonicity of negation as failure, which enables easy modelling of interaction among general and specific rules. Answer set programming (ASP) implements stable model semantics. Due to modern, highly optimized grounders and SAT solvers, ASP implementations are fast enough for many applications and are mostly used for high-level reasoning tasks such as planning, diagnostics, learning and scheduling [Gelfond 2008]. This framework also looks very promising for NLP applications [Balduccini 2013], and especially for our problem. The leading ASP modelling language, Potassco, which was developed at the University of Potsdam [Gebser at al 2011], includes support for weak constraint and optimization. This enables the formalization of the sequence labelling task as an optimization problem and, therefore, seems particularly promising in the context of the joint inference model. 26

Dataset

The test set for the models consisted of 1774 transcribed records from the census, including name expressions, gender and nobility labels. The available dataset of 4018 labelled modern international names (12,075 tokens) was used for the initial training and test dataset. 27

Name parsing models

In order to evaluate and compare the appropriateness of the probabilistic and rule-based models for the task, both conditional random field (CRF) and rule-based models based on stable model semantics (SM Rules) were developed. They shared tag set and features. The selection of a tag set is an important task, since it influences the accuracy of the learned models and the usability of the model. If there are fewer numbers of categories, the accuracy will generally improve, but less structure will be introduced and less ambiguity will be reduced. 28

For a tag set, the following classes were selected: 29

- **N.FN.(M/F)**: male/female first name; e.g., “Gustav/ Josephine”
- **N.LN**last name, e.g., “Philippovich”
- **N.LN.PREF**last name prefix, e.g. “de,” “von”
- **N.TITLE**: person title, e.g., “pl.” (noble), “dr.”
- **N.QUAL**: surname qualification, e.g., “ml” (junior)
- **N.SALUT**person salutation, e.g., “herr” (mister)
- **GEO**:geographic/location term, e.g., “Zagreb,” “Ilica”
- **OTHER**, terms not in the above list, like notes, comments, etc.

All tags, except OTHER, had standard –B and –I suffixes, for denoting multi-token tags.

The features for both models included token, n-grams, packed word forms (lower/upper-case combination sequence, packed to three characters), and a dictionary entry from an available international name dictionary, including estimation of monogram frequency. 30

The CRF model was trained in the standard way, using the CRFsuite [Okazaki 2007]. 31

Rule based-system

The rule-based model was implemented in the Potassco ASP system [Leuschel and Schrijvers 2014.]. 32

Head of the rules were in the following form: `tag(I,P,[tag],[weight], [level])` where `I` is id of record, `P` is 33

position of token in name expression, `[tag]` is one of the tags from the tag set, `[weight]` is an estimation of certainty of the rule, and `[level]` represents generality of the rule, as explained below. Rule body is a set of (possible negated) features. An example of this rule is:

```
tag(I,P,n_title_b,70,1) :- lexc(I,P,n_title_b,_,_),
wordform(I,P,"LlLlLl"), wordform(I,P-1,"LuLlLl"), lexc2(I,P-1,n_fn,_,1),
lexc2(I,P+1,n_ln,_,1), not specExists(I,P,1).
```

It defines that token at position `P` is a title if it has a dictionary entry for title at any frequency, it is lowercase ("LlLlLl" wordform), token before it is capitalized and the most frequently used first name, and the token after is the most frequently used last name according to the dictionary.

The last atom in the rule (`specExists`) used the non-monotonic nature of stable model semantics, stating that the rule is satisfied only if there is no other rule for the same token that is more specific. The most general (default) rules were on level 1, more specific on level 2, etc. 34

Some of the initial rules were hand-coded, but the majority of them were learned from the initial training dataset. Although much research has been performed on rule induction [Muggleton 2015], there is no suitable rule learning system available for ASP; as such, a rudimentary one was developed, inspired by Inductive logic programming algorithms. 35

Learning rules

The pseudo code for the preliminary rule induction system was as follows. 36

```
Generate all features of examples in the training set
Generalize features [replace constants with variables, relativize positions]
Select top-n features (eliminate all with low chi-square in the training set)
for lev in 1 to maxLevel
        predicted = tag training set with rules up to level lev-1
        for tag in tag-set
                for x in power set of features up to length maxCardinality
                        gain = count false negative matching x in predicted
                        loss = count true negative matching x in predicted
                        if gain>loss add x to rules candidates
                for y in rules candidates sorted by gain-loss
                        if rules does not overlap with rules add y to rules
```

The hyper-parameters `maxLevel` and `maxCardinality` control the number of level of specific rules and the number of atoms in rules. For performance reasons, the learning system was implemented in Python and Potassco ASP. Trained on the modern language training set, the system generated 218 rules on four levels of generality. Token level f1-score, measured on 20% of the modern data-set was 0.95. 37

Results

We were interested to see whether the model could correctly classify all the parts of a record; i.e., name expressions. Therefore, instead of the more common precision/recall/f-score measure of token level classification accuracy, only the items where all tokens were correctly classified were counted as correct. Therefore, the parsing accuracy was defined as the percentage of test records for which the test results were identical to the manually parsed records. 38

Statistical and rule-based models were evaluated as trained on initial modern language training set and after improving models. The SM Rules model was improved by hand-writing four additional rules that were obvious from the errors in 39

the first model. The CRF model was improved by labelling 100 additional records from the source and adding these to the dataset. This task took approximately twice the time it took to write the rules. This can be considered to represent a similar investment of resources, although it is not a precise measure of the effort invested in improving the models because both procedures depended on the characteristics of the datasets and the experience of the researcher.

	Initial model		Improved model	
	SM rules	**CRF**	**SM rules**	**CRF**
Accuracy rate	79.82%	67.93%	97.01%	76.21%
Support	1416	1205	1721	1352

Table 1. Parsing evaluation

As the data in the table above clearly indicate, the models were significantly different, with the p-value of McNemar test for both being `lt 2.2e-16`. 40

Gender and nobility model results

The gender and nobility prediction model was based on character n-Gram (length 1-9) of name expressions in census data. Name expressions were pre-processed by marking the beginning and end, lowercasing and stripping accents. A support vector classifier with linear kernel was used, and the parameters were obtained by grid search. As the results of k-folding cross validation were satisfactory, the initial plan for building rule-based classifiers was abandoned. 41

Nobility status prediction					Gender prediction				
Class	**precision**	**recall**	**f1-score**	**support**	**Class**	**precision**	**recall**	**f1-score**	**support**
Noble	0.98	1.0	0.99	321	Male	0.98	0.99	0.98	192
Common	1.0	0.85	0.92	34	Female	0.99	0.98	0.98	163

Table 2. Gender and nobility evaluation table

Discussion

Summary of the researchers' experience in applying statistical and rule-based approach to historical text is given in the following table: 42

CRF		Rule-based	
Drawbacks	**Advantages**	**Drawbacks**	**Advantages**
CRFs must be trained on a new training set whenever a historical source is systematically different from a previously built model.	CRFs are widely used, so it is easy to use implementations of CRF models.	If rules are hand-coded, it has to be done by researchers, trained and experienced in both domain specific knowledge and a rule-based system.	Possibility of coding general and domain specific constraints and rules.
Models are next to impossible to be modified ad hoc, in order to explore observed regularities in a new domain or historical source.	Outperforms other models (including HMM) in many application domains.	Learning algorithms are inferior to the ones used to train statistical models.	Learning can be performed on top of the hand-coded rules.
In semantically opaque models, there is no easy understandable answer to a “why” question.	Models can be developed from dataset labelled by persons lacking linguistic and/or computer science skills.	Complex interaction of rules can make it difficult to understand and modify the rules ad hoc.	Resulting rules are relatively semantically transparent and can be modified and improved ad hoc.

Table 3. Comparison between the statistical and the rule-based approach

Conclusion and Outlook

The preliminary results indicated that the rule-based approach, which was based on stable model semantics, is more suitable for inferring standard name forms from historical texts than the more widespread statistical approach. To confirm this result, the experiment should be repeated using additional historical sources and statistical models. To predict gender and nobility, it seems more convenient to use standard statistical classifiers when labelled data is available. The generalization accuracy of the models should be tested on additional historical sources. A model ensemble that includes both a rule-based method and the CRF model is another interesting development that is worth a future research. 43

In order to make the model more suitable for real-world applications in historiographical research, it would be worthwhile to develop an interactive interface that would enable incremental rule learning. It should use a simple web interface and the rule induction system should recommend source-specific rules to the researcher, hiding the underlying complexity of the rule system. 44

The development of a more complex system that includes joint inference from the scan of a source to a historical demography web ontology is a worthwhile longer-term goal. This research represents a small step toward the development of such a system. 45

Notes

[1] Although the exact boundary between structured and unstructured data is imprecise, for the scope of this research we can define structured data as those data in which ambiguity is reduced to the level where no additional human intervention is needed to perform desired processing. Ambiguity exists on different levels. In the analysis of text, one commonly distinguishes between lexical and structural ambiguity. However, in the context of analysing historical documents, semantic ambiguity is important. Even if we have, for example, a name of the person or a place in proper digital format, ambiguity of whether this person is the same one as in previous document makes this information unstructured for some purposes. Similarly, if the word “father” occurs in the text, it is important to distinguish whether this is a binary predicate (a relation between two individuals) or a singular predicate (property of one individual being a priest). In the context of our use case, if we have a list of only full names of persons in database and we want to make mailing labels for them, one can say that they are structured; but if we want to list them by surnames, they are unstructured because that cannot be performed easily.

[2] The notable exception is the Baška tablet, written in Glagolitic script. It is one of the first monuments containing an inscription in the Croatian recension of the Church Slavonic language, dating from c. 1100. However, most of the historic documents were written in the Latin language, mostly within diplomatic collections.

Works Cited

Balduccini 2013 Balduccini, M.“Some Recent Advances in Answer Set Programming (from the Perspective of NLP),” *2013 CEUR Workshop Proceedings*. 1044. 1-6.

Boren 2007 Boren, L., e. a., “Naming the Past: Named Entity and Animacy Recognition in 19th Century Swedish Literature.” In: *ACL 2007. Proceedings of the Workshop on Language Technology for Cultural Heritage Data.* (2007) pp. 1-9.

Brill 1992 Brill, E., “A simple rule-based part of speech tagger.” Stroudsburg, PA, USA, Association for Computational Linguistics, (1992) pp. 152-155.

Chiticariu 2010 Chiticariu, L. a. a., 2010. “Domain adaptation of rule-based annotators for named-entity recognition tasks.” Massachusetts, USA, Association for Computational Linguistics , (2010) pp. 1002-1002.

De Wilde 2017 De Wilde, M., “Semantic Enrichment of a Multilingual Archive with Linked Open Data.” *Digital Humanities Quarterly* Vol 11/4 (2017).

Devlin at al 2018 Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., “Bert: Pre-training of deep bidirectional transformers for language understanding”. (2018) arXiv preprint arXiv:1810.04805.

Foka 2018 Foka, A., “Digital Technology in the Study of the Past.” *Digital Humanities Quarterly* Vol 12/2 (2018).

Gebser at al 2011 Gebser, M. & all, a., “Potassco: The Potsdam Answer Set Solving Collection.” *AI Communications - Answer Set Programming*, (2011) pp. 107-124.

Gelfond 1998 Gelfond, M. L. V., “The stable model semantics for logic programming. ” (1998) p. 1070–1080.

Gelfond 2008 Gelfond, M., 2008. “Answer sets.” In: *Handbook of Knowledge Representation*. s.l.:Elsevier, p. 285–316.

Graves 2012 Graves, A., “Supervised Sequence Labelling.” In: *Supervised Sequence Labelling with Recurrent Neural Networks*. s.l.:Springer Berlin Heidelberg, (2012) pp. 5-13.

Heckmann at all 2014 Heckmann, D., e. a., “Citation segmentation from sparse & noisy data: A joint inference approach with Markov logic networks.” *Digital Scholarship in the Humanities* (2014) pp. Vol 31/2. 333-356.

Johannessen at al 2005 Johannessen, J. B., e. a., “Nemed Entity Recognition for the Mainland Scandinavian Languages.” *Digital Scholarshipin the Humanities*, pp. (2005) Vol 20/1. 91-102.

Lauc and Vitek 2018 Lauc, D., Vitek, D. “From the History to the Story: Harvesting Non-Monotonic Logic and Deep Learning to Generate Multilingual Family Narratives from Genealogical Data.” DH Budapest 2018 Conference. (2018)

Leuschel and Schrijvers 2014. Leuschel, M. & Schrijvers, T., “Technical Communications of the Thirtieth International Conference on Logic Programming (ICLP'14).” s.l., s.n. (2104)

McCallum 2009 McCallum, A., “Joint inference for natural language processing.” Boulder, Colorado, Association for Computational Linguistics. (2009)

Muggleton 2015 Muggleton, S. H. W. a. H. W., “Latest Advances in Inductive Logic Programming.” s.l.:Imperial College Press. (2015)

Ndeau and Sekine 2007 Nadeau, D. & Sekine, S., “A survey of named entity recognition and classification.” “Lingvisticae Investigationes.” (2007)

Okazaki 2007 Okazaki, N., “CRFsuite: a fast implementation of Conditional Random Fields (CRFs),” s.l.: s.n. (2007)

Osborne 2000 Osborne, M.“Shallow parsing as part-of-speech tagging.” s.l., ACM, (2000) pp. 145-147.

Petran 2012 Petran, F., “Studies for Segmentation of Historical Texts: Sentences or Chunks?. On Annotation of Corpora for Research in the Humanities” ACRH-2, (2012) p.75.

Poon and Domingos 2007 Poon, H. & Domingos, P., “Joint inference in information extraction.” *AAAI*m (2007) pp. 913-918.

Sha and Pereira 2003 Sha, F. & Pereira, F., “Shallow parsing with conditional random fields.” Edmonton, Canada, Association for Computational Linguistics, (2003) pp. 134-141.

Vamvakas 2008 Vamvakas, G. e. a., “A complete optical character recognition methodology for historical documents.” (2008) IEEE.

Viet Cuong at al 2014 Viet Cuong, N. a. a., “Conditional random field with high-order dependencies for sequence labeling and segmentation.” *The Journal of Machine Learning Research* Vol 15/1, (2014) pp. 981-1009.

Vrbanus 2010 Vrbanus, M. “Skrivena povijest – tajnoviti svijet brojki.” Povijesni prilozi. Vol. 39/39., (2010) pp. 39-71.