Iterative End-to-end Information Extraction based on Linear Models

Slavko Žitnik[1,2], **Lovro Šubelj**[1], **Marko Janković**[1], **Bojan Furlan**[3], **Dražen Drašković**[3], **Nemanja Kojić**[3], **Marko Mišić**[3], **Marko Bajec**[1]

[1] *University of Ljubljana, Tržaška cesta 25, SI-1000 Ljubljana*
[2] *Optilab d.o.o., Župančičeva 8, SI-5270 Ajdovščina*
[3] *University of Belgrade, Bulevar kralja Aleksandra 73, RS-11120 Belgrade*
{*slavko.zitnik, lovro.subelj, marko.jankovic*}*@fri.uni-lj.si*, {*bojan.furlan, drazen.draskovic, nemanja.kojic, marko.misic*}*@etf.bg.ac.rs*, *marko.bajec@fri.uni-lj.si*

Information Extraction is a process of extracting structured data from unstructured sources. It roughly consists of tasks like entity extraction, relation extraction and coreference resolution. Most of the current research focuses only on one of the tasks or their combination in a pipeline. In this paper we introduce an end-to-end iterative information extraction system. We propose a novel dataset tranformation which enables the use linear-chain conditional random fields for all the three tasks. The usage of efficient linear model enables faster training and inference with parallelization possibilities.

1 Introduction

The mining, extraction and integration of useful data, information and knowledge from unstructured sources, is however far from easy. This specifically holds for the *information extraction* (IE), which has been addressed by the research community for a long time already. The objective of the IE is to gather structured data from unstructured sources (e.g., text documents, images, videos). The IE can be further divided into several tasks, three of them being most important: (1) *entity extraction* (EE) that deals with the identification of entities, (2) *relation extraction* (RE) dealing with the identification of relationships between entities, and (3) *coreference resolution* (COR) that identifies phrases corresponding to the same entity.

While the early IE systems (can be traced back to early 90's) were mostly naive and rule based, later (semi-)automatic techniques of wrapper generation, seed expansion or rule induction were developed [11]. Further on, the machine learning approaches gained popularity and outperformed other approaches on general datasets. The most promising results have been however achieved by different graphical models, especially *conditional random fields* [2] (CRF). In contrast to traditional machine learning classification or regression techniques, CRF as sequence tagger benefits from the representation of sentences as the sequence of words.

In this paper we propose a new ontology-based IE system that combines the tasks of EE, RE and COR in an iterative method. For all the subtasks we use (linear-chain) CRF models for training and inference. We use the ontology for the process guidance, domain, additional rules for the *feature functions*, and a schema for semantic database. Furthermore, we introduce a special dataset transformation which enables the usage of only linear models and the use of parallelization.

The rest of the paper is organized as follows. Section 2 gives a brief review of the related work. Next, the CRF is presented and the iterative ontology-based information extraction system with dataset transformations is proposed. In Section 4.1 we describe system's general training and inference algorithm. Lastly we discuss the proposed solution and reveal further work.

2 Related work

Early work in IE was mostly driven by shared efforts at MUC (*Message Understaning Conference*) and CoNLL (*Conference on Computational Language Learning*), and by the ACE (*Automatic Content Extraction*) program. Since the vast majority of the research on each of the IE tasks was done independently [11], the applicability of the proposed approaches was somewhat limited. Nevertheless, the most thoroughly investigated task EE is relatively well solved, with state-of-the-art approaches achieving accuracy over 90% on general datasets [8]. In contrast to the latter, RE and COR approaches achieve only up to about 70% [4, 3].

The idea of an *iterative IE* (also *collective IE*) was first employed for EE by exploiting mutual influence between possible extractions [1]. In [6], authors proposed an iterative system combining EE and RE with knowledge integration from an ontology. Although the system is rule-based, it was an important step towards a general approach. Felix [7], a state-of-the-art IE system based on *Markov logic networks* [10], accepts generally applicable rules and scales to very large datasets. The system has been tested against EE and COR with promising results, still, it requires a substantial amount of manual input.

Ontology-based IE has recently emerged as an important subfield of IE [5]. Ontologies represent an additional knowledge that can be efficiently employed during the extraction process. Most modern systems use a single ontology for domain representation [9], however, there is no rule against using more of them.

The main contributions of this paper are as follows: (1) generic ontology-based IE architecture (adaptable to arbitrary domain or language), (2) a holistic IE approach (combining all three main tasks and intermediate results), (3) faster training and inference due to an efficient CRF model (straightforward adaptation to parallel architecture), and (4) the special dataset transformations to enable EE, RE and COR using simple linear models.

3 Conditional random fields

Conditional random fields (CRF) [2] is an example of a discriminative model that estimates the joint distribution $p(\overline{y}|\overline{x})$ over the target sequence $\overline{y}$ conditioned on the observed sequence $\overline{x}$. For example, an observed sequence $\overline{x}_1$ is a sequence of words within the first sentence. Next to this there are also corresponding sequences that contain part-of-speech-tags, lemmas, parse trees, etc., respectively. These are used by different feature functions f_i employed by a CRF in order to model the target sequence $\overline{y}_1$. In our system we will predict three types of target sequences, one for each of the IE tasks (i.e., EE, RE and COR).

Training CRF actually means finding a weight vector w that predicts best possible (i.e., most probable) sequence $\hat{y}$ given $\overline{x}$. Hence,

$$\hat{y} = \arg\max_{\overline{y}} p(\overline{y}|\overline{x}, w), \tag{1}$$

where the conditional distribution equals

$$p(\overline{y}|\overline{x}, w) = \frac{\exp(\sum_{j=1}^{m} w_j \sum_{i=1}^{n} f_j(\overline{y}, \overline{x}, i))}{C(\overline{x}, w)}. \tag{2}$$

Here n is the length of the observed sequence $\overline{x}$, m is the number of feature functions and $C(\overline{x}, w)$ is a normalization constant computed over all possible $\overline{y}$. It can be omitted because we only take the most probable sequence and are not interested into the exact probabilityl.

The structure of a CRF is defined by the references to target sequence labels within the feature functions. A linear-chain CRF (LCRF) feature function calculated at position i can depend only on the current and the previous sequence labels y_i and y_{i-1}. For arbitrary structured CRF exact inference of weights is intractable due to an exponential number of partial sequences. Thus, approximate approaches must be adopted. On the other hand, the maximal length of a partial sequence in LCRF is limited to two, while training and inference can be easily and fast solved using forward–backward method and Viterbi algorithm. Note that LCRF have already been succesfully used for IE, especially at EE task.

3.1 Feature functions

The modelling of feature functions is the main source of increase of precision and recall when training CRF classifiers. Usually these are given as templates and final features are generated by scanning the entire training data. An example of a simple feature function can return 1 if the previous word is "Mr." and current word is capitalized, otherwise returns 0.

We categorise the feature functions into four categories: Preprocessing, String, Semantic and Iterative. Detailed description of a complete set of feature functions is ommitted due to space limitations. (Exact feature functions we use can be retrieved from the system's source code.)

4 Iterative IE system

In the following section we give a high-level description of the proposed iterative end-to-end IE system. Then we present general training and inference methods for all three subtasks. Source repository of the whole proposed system with relationship extraction and coreference resolution evaluation is publicly available (`https://bitbucket.org/szitnik/iobie`).

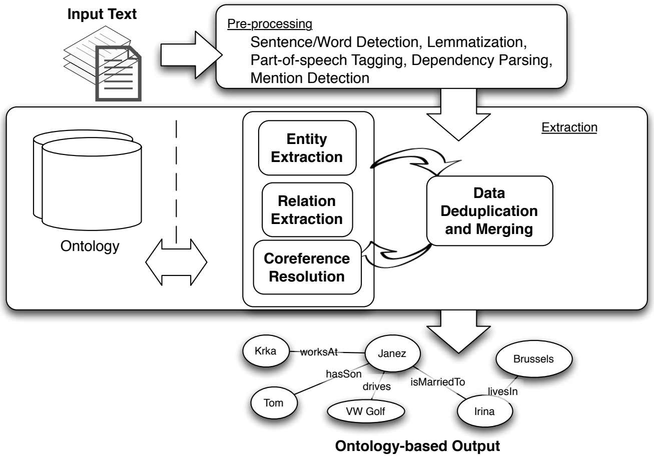

Figure 1: The proposed iterative end-to-end information extraction system.

The proposed high level architecture of the end-to-end IE system is shown in Fig. 1. The term end-to-end is used because the system takes raw text as input and returns fully processed result for all three IE subtasks. The main modules are the following:

(1) Preprocessing module enriches the input data with additional attributes required in the subsequent modules. In particular, the module detects sentence and word boundaries, lemmatizes the words, performs part-of-speech tagging, dependency parsing and other (details are omitted due to space limitations). Note that this is the only part of the system that is language dependent. (When no preprocessing methods are available for a certain language, the module must at least identify sentence and word boundaries.)

(2) Extraction is the system's most important module that consists of an iterative approach for EE, RE and COR combined with data deduplication and merging techniques. Training and inference of the underlying CRF models is outlined in Section 4.1. Since CRF models are used for EE, RE and COR, annotations from entity, coreference and relation tasks can be easily transfered throughout the system. Data deduplication and merging connects identified entities and their coreferences into a semantic graph via extracted relationships. For this task we use collective entity resolution techniques. Note that the resulting graph enables the identification of pairs of distant entities or coreferences, while it also represents an input for additional feature functions at the next iteration of the extraction. The extraction proceeds until no change is detected in two consecutive iterations, or when the maximum number of iterations is reached.

(3) Ontology module is used in three different contexts. Firstly, the ontology represents the underlying *domain* modelled by EE and RE tasks (i.e., entities and relations are represented as ontology concepts and properties). Secondly, the ontology can also define arbitrary *concepts, constraints or rules* (e.g., distance between concepts, neighborhood of a concept, regular expression that a concept must conform to etc.). These are used directly by feature functions, thus system performance can be improved by ontology population. Note that this is the only part of the system that can be manipulated by the user. Thirdly, ontology also serves as a *data store schema* for extracted entities like a *gazetteer list* (i.e., a set of known instances) per each concept.

Output of the system consists of a semantically annotated graph that is, with the user's consent, saved into the internal semantic data store.

4.1 Training and inference

The proposed architecture takes raw text as input and returns semantically annotated text as output. Because we treat all three subtasks EE, REL and COR as sequence labelling tasks, we transform the input data into a unified representation during the preprocessing step.

Let $\overline{x} = [x_1, x_2, ..., x_n]$ denote a sequence of mentions from one document. Mention is every text reference that refers to a specific named entity. Our goal is to predict the target sequence $\overline{y}^{l_i}$, where $l_i \in \{EE, REL, COR\}$. For EE we use standard labelling as is traditionally proposed in the field. For RE we label i-th mention with the name of a relationship which exists between mentions x_{i-1} and x_i or otherwise with `O` (i.e. Other). For COR we label i-th mention with `C` if mentions x_{i-1} and x_i are coreferent or otherwise with `O`.

To enable the extraction of relationships and coreferences over larger distances (i.e. having one or more mentions in between) and still using linear models, we propose a special skip-mention sequence transformation. An example of transformation from initial to two other skip-mention sequences for the COR task is shown in Fig. 2. We call initial sequence (e.g. $\overline{x}$) zero skip-mention sequence, sequence that contains every second mention is one skip-mention sequence, etc.. For each skip-mention distance we train separate LCRF model and then combine the results of all models. Training and inference can thus be parallelized.

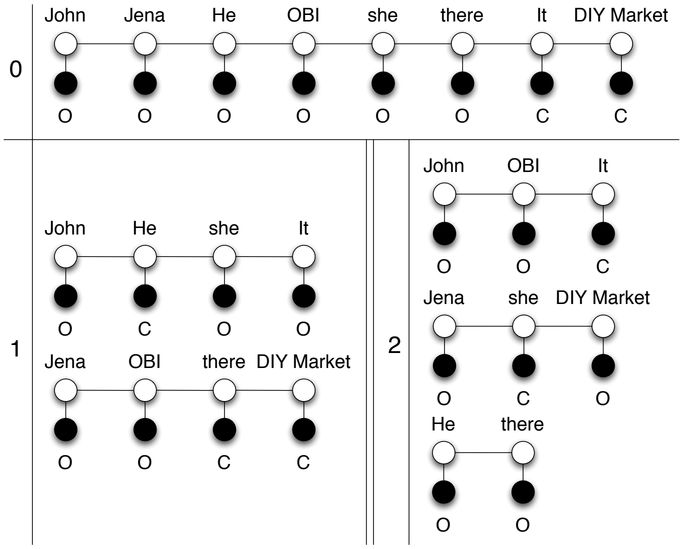

Figure 2: Transformation into skip-mention sequences for input sentence "John is married to Jena. He is a mechanic at OBI and she works there. It is a DIY market.".

The results show that we can identify relationships with high accuracy [12]. Preliminary results for the COR task also show similar improvements over traditional methods.

We show a high level implementation of one iteration for one of the subtasks over skip-mention sequences in Fig. 3. We first detect mentions from input documents [3]. From these we form multiple sequences and perform model loading. When training, where we already have tagged data, the new models are learned and then loaded. Using these models we perform inference and combine the results using entity resolution techniques. The iterative method is repeating these steps for each of the EE, RE and COR tasks. Lastly, entities (i.e. clusters of mentions) with extracted rela-

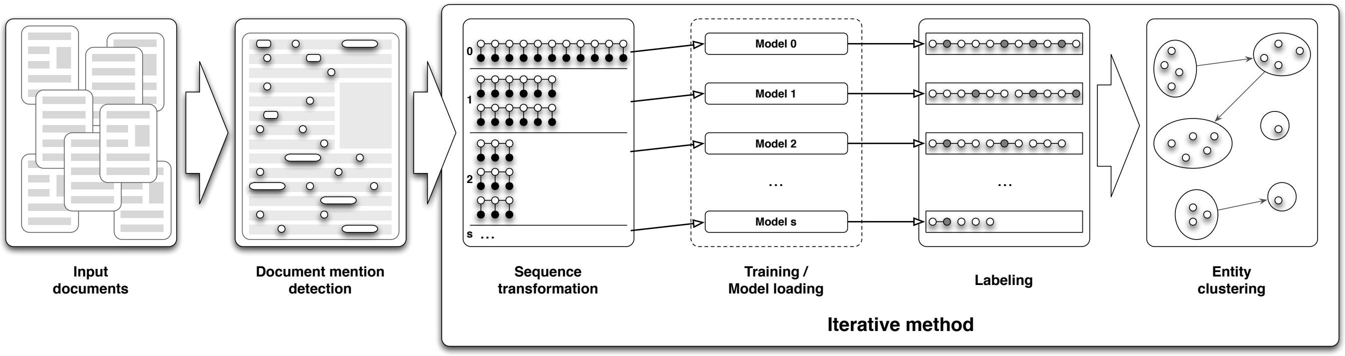

Figure 3: High level skip-mention information extraction architecture.

tionships are returned as a final result.

We tested the system only using EE and RE subtasks and achieved promising results [13].

5 Conclusion

This paper proposes an iterative end-to-end information extraction system that uses linear-chain conditional random fields only. The system employs three main tasks - entity extraction, relation extraction and coreference resolution with additional labelling transformations. We categorise feature functions and present new iterative ones to take into account intermediate labellings from previous iterations and semantic ones, which are not only for guidance, but also for domain, rules and semantic database schema definition.

6 Acknowledgements

The work has been supported by the Slovene Research Agency ARRS within the research program P2-0359 and part financed by the European Union, European Social Fund.

References

[1] R. Bunescu and R. J. Mooney. Collective information extraction with relational markov networks. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, 2004.

[2] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, 2001.

[3] H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the CoNLL-2011 Shared Task*, 2011.

[4] Y. Li, J. Jiang, H. Chieu, and K. Chai. Extracting relation descriptors with conditional random fields. pages 392–400, Thailand, 2011. Asian Federation of Natural Language Processing.

[5] L. McDowell and M. Cafarella. Ontology-driven information extraction with ontosyphon. *The Semantic Web-ISWC 2006*, page 428–444, 2006.

[6] C. Nedellec and A. Nazarenko. Ontologies and information extraction. *CoRR*, abs/cs/0609137, 2006.

[7] F. Niu, C. Zhang, C. Ré, and J. W. Shavlik. Felix: Scaling inference for markov logic with an operator-based approach. *CoRR*, 2011.

[8] J. Nothman, N. Ringland, W. Radford, T. Murphy, and J. R. Curran. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, 2012.

[9] B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, and A. Kirilov. Kim – a semantic platform for information extraction and retrieval. *Natural Language Engineering*, 10(3-4):375–392, Sept. 2004.

[10] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62:107–136, 2006.

[11] S. Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3):261–377, 2008.

[12] S. Zitnik, M. Žitnik, B. Zupan, and M. Bajec. Extracting gene regulation networks using linear-chain conditional random fields and rules. In *Proceedings of BioNLP Shared Task 2013 Workshop*, Sofia, Bulgaria, Aug. 2013. Association for Computational Linguistics.

[13] S. Žitnik, L. Šubelj, D. Lavbič, A. Zrnec, and M. Bajec. Collective information extraction using first-order probabilistic models. In *BCI 2012: proceedings*, page 279–282, 2012.