Iterative joint extraction of entities, relationships and coreferences from text sources

Slavko Žitnik and Marko Bajec
University of Ljubljana
Faculty of computer and information science
Večna pot 113, SI-1000 Ljubljana
{slavko.zitnik, marko.bajec}@fri.uni-lj.si

***Abstract*—Machine understanding of textual documents has been challenging since the early computer era. Since the information extraction research field emerged it has inferred multiple natural language processing tasks, such as named entities recognition, relationships extraction and coreference resolution. Even though for the purpose of the end-to-end information extraction all of the three tasks are crucial, existing work has been focusing merely on one specific task at the time or at best on their connection in a pipeline. In this paper we introduce a novel iterative and joint information extraction system that interconnects all the three tasks together using iterative feature functions which use the advantage of the intermediate extractions. Furthermore, we introduce a special transformation of data into skip-mention sequences to enable the extraction of relations and coreferences using fast first-order graphical models. Additionally, the system uses an ontology as its knowledge source, as a list of inferred extraction rules, and as a data schema of extracted results. Experimental results show that the accuracy of extractions improves after each iteration. In particular, our model obtained a 15% error reduction on named entity recognition over individual models.**

I. Introduction

Information extraction (IE) gained importance in the 1970s, when early systems were focused mostly on the automatic detection of named entities in textual data [1]. Since then, a large number of IE systems dealing with entity extraction, relation extraction, and/or coreference resolution tasks have been proposed in the literature [2], along with the latest based on ontologies [3]. Information extraction [2] thus attempts to analyze text and extract its structured semantic contents. The extracted results therefore enable new ways to query, organize, analyze or visualize data. These type of information systems thus ease web searching by the use of structured data, automatically extract opinions, structurally compare products from unstructured reviews, etc. Recently, the same techniques were adopted in bioinformatics field to extract biological objects (e.g. proteins, genes), their interactions and experiment results from the vast biomedical databases [4]. Information extraction techniques have roots in the natural language processing community, as text was one of the first and still is highly important unstructured information source in the field. Nevertheless, the term is also used to extract structured data from arbitrary source types such as, videos, images or sounds.

The most important information extraction tasks consist of named entity recognition or entity tagging, relationship extraction and coreference resolution (i.e., clustering of mentions to an entity). Prior to employing these tasks, input data needs to be preprocessed. During the data preprocessing, we transform the input into the appropriate data representation and enrich it with additional data (e.g., lemmas, part of speech tags) that improves the whole information extraction. The entity tagging task takes a sentence of words or symbols (i.e., tokens) as an input and detects the entity type for each token, which can be, for example, a person, a location or an organization. The relationship extraction identifies relationships (e.g., works at, is a) between text phrases (Figure 1). These phrases are attributes of a relationship and are called mentions. The coreference resolution task [5] is the task of detecting mentions in the text that refer to the same underlying entity [6] (i.e., the subject of the discussion that is then digressed, changed, etc.). Mentions can be of either named (e.g., "John Doe"), nominal (e.g., the guy with the glasses), or pronominal type (e.g., he or him) [7]. The goal of coreference resolution is thus to detect groups of mentions that refer to the same real-world entities. To accomplish this, one employs, apart from an initial text preprocessing, mention detection (i.e., identification of phrases that represent valid entity mentions), and mention clustering (i.e., determining which pairs of mentions corefer). Since the former can be solved in a rather straightforward fashion [8], we here consider only the last (we assume that the mentions in the text are given). In Figure 2 we show an example of an end-to-end information extraction using all the three tasks in a text document.

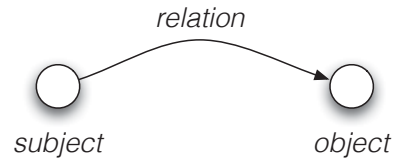

Fig. 1. **General relation representation.** Each relation (e.g., Jena works at OBI) is defined with a name (e.g. *worksAt*) and subject (e.g. *Jena*) and object (e.g. *OBI*) relationship attributes.

The early information extraction research was strongly driven by Message Understanding Conference (MUC) competitions from 1987 (MUC-1) to 1997 (MUC-7). Initial challenges focused merely on named entity recognition. Later, important competitions supporting more tasks and containing larger data corpuses emerged, like Automatic Content Extraction (ACE) [9], Semantic Evaluation (SemEval) [10] and Conference on Natural Language Learning (CoNLL) Shared Tasks [11]. In the biomedical field the BioCreative challenges I and II focused only on the detection of protein/gene mentions

978-1-4673-6630-4/15/$31.00

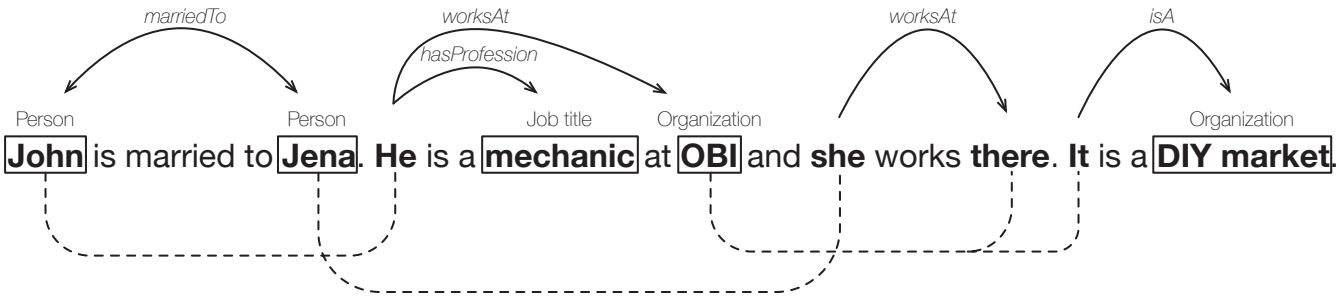

Fig. 2. **Information extraction.** Representation of the main information extraction tasks named entity recognition (e.g., person, job title, organization), relationship extraction (e.g., married to, works at) and coreference resolution (i.e., dashed line connecting mentions that represent the same underlying entity).

[12]. Furthermore, the LLL [13] and BioNLP [4] challenges addressed other information extraction tasks, such as coreference resolution and relation extraction.

In this paper we propose an iterative joint system for end-to-end information extraction which combines named entity recognition, relationship extraction, and coreference resolution. We expect that by taking into account intermediate extractions from other tasks will improve the overall performance of the system in each iteration. We also propose novel transformations of data into skip-mention sequences so that we can model all of the three tasks with linear-chain conditional random fields model. The model is traditionally successful for named entity recognition. It offers fast training and inference with great support for a lot of features. Furthermore, we evaluate the proposed system on ACE 2004 dataset, achieve comparable results to the state-of-the art systems, detect minor improvements over the three iterations, and achieve a 15% of error reduction for named entity recognition.

The rest of the paper is structured as follows. In the following section we present an overview of the related work for all the three information extraction tasks and previous work on joint data extraction. Next we introduce the conditional random fields algorithm with basics of sequence labeling and feature functions definition. In Section 4 we present our proposed system for iterative joint information extraction. First, we introduce an appropriate data representation and show the motivation for the selected approach for all the three tasks. This makes the extraction possible using linear-chain conditional random fields. Second, we explain the whole system with the system ontology and categorize feature functions that we use. In Section 5 we show the results of the evaluation of the proposed system, then discuss the results, and then provide a conclusion.

II. Related work

A majority of research in information extraction focuses on individual tasks or on a connection of them into a simple pipeline [2]. The latest results show that the named entity recognition is solved quite well, achieving around 90% or more F-score on general data sets, while coreference resolution achieves about 70%, and relationship extraction around 50%.

Named entity recognition is one of the first researched tasks with a variety of different approaches proposed. Traditional ones are roughly classified as pattern-based and machine-learning-based. The first ones extract entities using some templates, dictionaries or predefined set of rules [14], [15], [16], while the latter employ machine learning classifiers and induction methods [17], [18] to label tokens with a predefined set of entities. The task is defined as a sequence labeling task and thus a lot of approaches use sequence classifiers such as, hidden Markov models, maximum entropy models or conditional random fields [19].

Relationship extraction systems generally categorize themselves into two categories. Feature based systems use a variety of lexical, syntactic and semantic features [20], [21]. The other common approach uses kernel methods [22], [23]. Some methods also cast the task as a sequence labeling task and tag text phrases (i.e., relationship descriptors) that represent a predefined relationship [24]. On the other hand, in the subfield of open information extraction some unsupervised approaches that extract arbitrary relationships were proposed [25], [26], but they do not classify them or their attributes.

The majority of techniques for coreference resolution transform the problem into a pairwise classification task [27], [28] (i.e., the algorithm checks every pair of mentions for coreference). This enables the use of standard machine learning classifiers. Thus, a number of innovative and linguistic-rich feature functions [28], [29], along with different algorithms like maximum entropy [30], SVM classifiers [31] and Markov Logic Networks [32], have been proposed in the recent literature. On the other hand, unsupervised techniques infer the coreferentiality based on sequences of mentions [8], [33], which are much harder to train and are not easily generalized to new problems or domains but achieve state-of-the-art results on known domains. McCallum et al. [34] was the first to propose the three general conditional random fields (CRF) models to solve the coreference resolution problem. The first is a general model (i.e., the CRF structure is unrestricted) and the training or inference is therefore complex. The second model represents pairs of mentions by specific attributes, while the third represents the pairs as nodes in the model. Wellner et al. [35] successfully applied coreference resolution to citation matching, interestingly by using a special case of McCallum's first model combined with named entity extraction. Due to the tractability issues of general models, an extension of skip-chain CRF has been proposed [36], which also supports the use of long-distance dependencies by incorporating additional cliques into the model. Cullota et al. [27] proposed the use of first-order probabilistic models over sets of mentions; thus, the algorithm operates directly on the entities.

Ontology-based information extraction has recently emerged as an important subfield of information extraction [3]. Ontologies represent an additional knowledge that can be efficiently employed during the extraction process [37], [38], [39]. Most modern systems use a single ontology for domain representation [40], however, there is no rule against using a combination of them.

In contrast to individual or pipeline-based approaches, the idea of iterative or joint information extraction by multiple subtasks tries to interconnect the tasks together with some mutual benefits [41]. It was first employed in named entity recognition by exploiting mutual influence between possible extractions [42]. Then some systems that jointly extracted entities and relationships using an ontology and rules were proposed [43]. Further, Yu and Lam [44] modeled the extractions of both using a discriminaive model. Roth and Yih [45] proposed the inductive logic programming framework to provide manual constraints between the tasks. Yao et al. [46] automatically inferred such rules through distant supervision via Wikipedia. The only work that is similar to ours in sense of jointly modeling all the three tasks was presented by Singh et al. [47]. They proposed a joint CRF model and an improved version of belief propagation to solve all three tasks with a single model. Other researchers have also tried to combine parsing with named entity recognition and semantic role labeling [48] or applied various information extraction tasks to citation segmentation and matching [49], [50], [35] and to other domains [51], [52].

III. Conditional random fields

Conditional random fields (CRF) [53] is a discriminative model that estimates the joint distribution $p(\overline{y}|\overline{x}, w)$ over the target sequence $\overline{y}$ conditioned on the observed sequence $\overline{x}$ and weight vector w (see below). We represent a sentence by a sequence of words x_i with additional corresponding sequences that represent attribute values such as, part-of-speech tags $x_i^{k_1}$, lemmas $x_i^{k_2}$, relationships $x_i^{k_3}$, and other observable values $x_i^{k_j}$. These values are used by feature functions f_l that are weighted during CRF training in order to model the target sequence $\overline{y}$. The sequence $\overline{y}$ corresponds to the source sequence and consists of the labels that we would like to automatically infer. For the named entity recognition target sequence we commonly use labels such as `PERSON`, `ORGANIZATION` or `LOCATION`. For relationship extraction we tag current token with a name of relationship (e.g., `WORKS AT`, `IS A`) if it is related to the previous token. For the coreference resolution task we tag a token with `C` if it is coreferent with the previous one, which is similar like in case of relationship extraction. In other cases, when there exist no entity, no relationship or no coreferentiality, we tag a token with `O`.

In the field of IE, CRFs have been successfully employed for various sequence labeling tasks and have achieved state-of-the-art results. They can also deal with a large number of multiple, overlapping, and non-independent features.

Training a CRF is thus maximizing the conditional log-likelihood of the training data, by which we find a weight vector w that predicts the most probable sequence $\hat{y}$ for given $\overline{x}$. Hence,

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

where the conditional distribution is

$$p(\overline{y}|\overline{x}, w) = \frac{\exp\left[\sum_{l=1}^{m} w_l \sum_{i=1}^{length(\overline{x})} f_l(\overline{y}, \overline{x}, i)\right]}{C(\overline{x}, w)} \tag{2}$$

Here, m is the number of feature functions and $C(\overline{x}, w)$ is a normalization constant computed over all possible sequences $\overline{y}$.

The structure of a CRF defines how the dependencies with target labels are modeled. A general graphical model (i.e., a graph denoting the conditional dependence structure) can depend on many labels and is therefore intractable for training or inference without complex approximation algorithms. Thus, we use only a simple linear-chain CRF (LCRF) model, which depends on the current and previous labels (i.e., a first order model). The structure of such a model is illustrated in Figure 3. Furthermore, with the use of a number of feature functions and special dataset transformations, our method achieves comparable results to the best known systems.

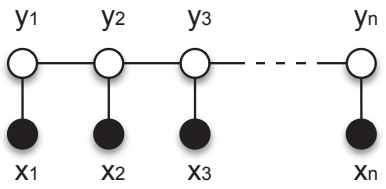

Fig. 3. **The structure of a linear-chain CRF model.** The model shows an observable sequence $\overline{x}$ (e.g., words) and a target sequence $\overline{y}$ (e.g., named entity tags, relationship tags, coreferentiality) containing n tokens.

Feature function modelling is an essential part of training a CRF. Selection of best feature functions may contribute to an increase in precision and recall when CRF classifiers are trained in a way that they achieve the highest level possible. Usually, these are given as templates and the final features are generated by scanning the entire training data set. The selection of informative features is the main source for an increase of precision and recall when training machine learning classifiers. Feature functions are usually implemented as templates and the final features are then generated by scanning the entire training data. In natural language processing, a few thousand or more features are commonly used, which can be efficiently handled by a CRF. A feature function that returns 1 if the current mention is of a person type or the previous mention is equal to "Mr." and 0 otherwise, is defined as follows:

$$f_l(\overline{y}, \overline{x}, i) = \begin{cases} 1, & \text{if } y_i = \text{PER} \vee x_{i-1} = \text{“Mr.”} \\ 0, & \text{if otherwise} \end{cases}$$

IV. Iterative joint information extraction

In this section we present our proposed system for iterative joint information extraction. First, we introduce novel data representation for all of the tasks along with extraction examples

using LCRF models. We then provide an overview of the end-to-end information extraction system that consists of input data preprocessing, iterative extractors that use an ontology with a data integration component which combines their results over multiple iterations.

A. Data representation

In the following subsections we introduce the data representation to enable the information extraction for all the three IE subtasks with LCRF models. For the named entity recognition we train the extraction models on a token-based sequences, while for relationship extraction and coreference resolution we use mention-based sequences. We represent mentions only as a subsequence of a token sequence and thus we work on the same data set with no special transformations for all three tasks.

1) Named entity recognition: Named entity recognition is the task of classifying tokens into the types of entities they refer to. We represent an input sequence $\overline{x}$ as a tokenized sentence, where each token is a word or other symbol. The result is a labeled target sequence $\overline{y}$ with predefined entity tags such as `PERSON`, `ORGANIZATION` or `LOCATION`. In Figure 4 we show a representation of the previous example (Figure 2) in our proposed system. To tag entities we follow traditional approach of tagging with first-order models, which we incorporate into an iterative method and extract entities jointly with relationship extraction and coreference resolution tasks.

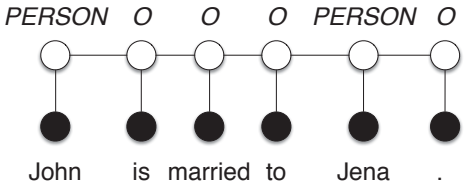

Fig. 4. **Named entity recognition input sequence.** The token-based sequence that is used by named entity recognition models for tagging with entity type tags.

2) Relationship extraction: The goal of the relationship extraction task is to identify relations between the two selected mentions. If we process the input sequences as is, we cannot model the dependencies between two consecutive mentions because there can be many other tokens in between. From the example (Figure 2) in the previous section we can observe the limitation of modeling just two consecutive tokens. With this type of labeling it is hard to extract the relationships using a first-order model. Also, we are not interested in identifying relation descriptors (i.e. segments of text that best describe a pre-defined relation); therefore, we generate new sequences containing only mentions. Mentions are also the only tokens that can stand as an attribute of a relation. In Figure 5 we show the transformation of our example into a mention sequence. The observable sequence $\overline{x}$ contains sorted entity mentions from a sentence that is annotated. These annotations are part of the training corpus.

The target sequence $\overline{y}$ is tagged with the none symbol (i.e. *O*) or the name of the relationship (e.g. *located, near, member of*). Each relationship target token represents a relationship between the current and the previous observable mention.

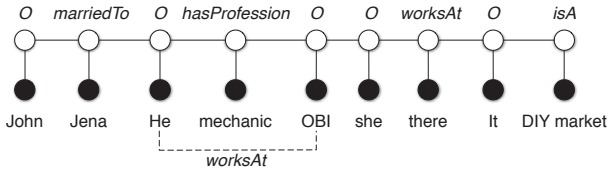

Fig. 5. **Zero skip-mention sequence for relationship extraction.** The initial mention sequence that contains all the mentions (i.e. zero skip-mention) from the example in Figure 2.

The mention sequence as demonstrated in Figure 5 does not model the relationships that exist between distant mentions. For example, the mentions *He* and *OBI* are related by a *works at* relationship, which cannot be identified using only a LCRF. A linear model can only detect dependencies between two consecutive mentions. To model such relationships on different distances, we generate appropriate skip-mention sequences. The notion of skip-mention stands for the number of other mentions between two consecutive mentions which are not included in a specific skip-mention sequence. Thus, to model relationships between every second mention, we generate two one skip-mention sequences for each sentence. A one skip-mention sequence identifies the *works at* relation, shown in Figure 6.

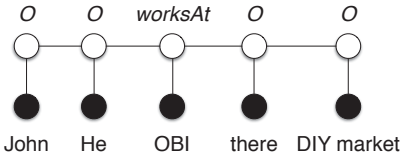

Fig. 6. **One skip-mention sequence for relationship extraction.** One out of two possible one skip-mention sequences, generated from the initial zero skip-mention sequence [*John, Jena, He, mechanic, OBI, she, there, It, DIY market*]. The other one consists of tokens *Jena, mechanic, she* and *It*.

Figure 7 shows the distribution of distances between the relation mention attributes (i.e. agents and targets) in the ACE 2004 training data set. From the distance distribution we observe that the majority of relations connect their attributes on distances of two mentions. It also shows the need to transform our data into skip-mention sequences. Without this transformation the linear-chain CRF model, at best, would only be able to uncover relations with attributes at zero distance (i.e. directly consecutive mentions).

3) Coreference resolution: The goal of coreference resolution task is to identify, which mentions corefer (i.e. represent the same underlying entity). Since we focus exclusively on the use of LCRF models only, we can identify only the coreferences over two directly consecutive mentions. Thus, to detect coreferences over larger distances, i.e., having one, two, three, or more mentions in between, we also propose a skip-mention data set transformation in the same way as for relationship extraction (see previous section).

For this task the observable sequence $\overline{x}$ denotes a sequence of all mentions within a document. Mentions x_i are ordered by their occurrence in the document. From our example document (Figure 2) we select all entity mentions into a zero-skip mention sequence $\overline{x} = [$`John`, `Jena`, `He`, `mechanic`, `OBI`, `she`, `there`, `It`, `DIY market`$]$ that is used for training.

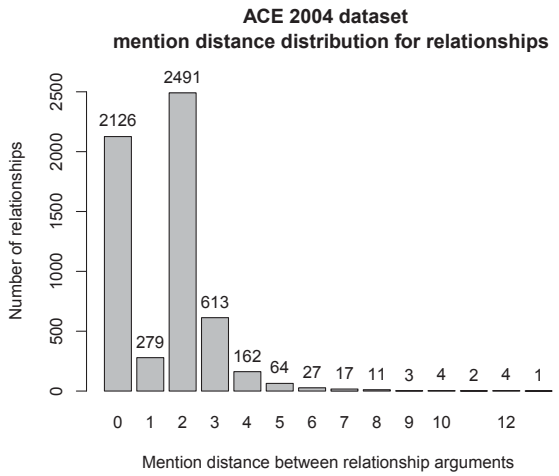

Fig. 7. **Distributions of distances between relationship attributes on ACE 2004 data set.** Distance x between two relationship mention attributes means that there exist x other mentions between them.

Our goal now is to detect the target clusters for each entity $x_{\texttt{John}} = \{\texttt{John}, \texttt{He}\}$, $x_{\texttt{Jena}} = \{\texttt{Jena}, \texttt{she}\}$, $x_{\texttt{OBI}} = \{\texttt{OBI}, \texttt{there}, \texttt{It}, \texttt{DIY market}\}$.

In some cases one mention could overlap with another mention. We treat such pairs as separate mentions and order them lexicographically by the index of the first word and mention length.

In Figure 8 we show a training mention sequence $\overline{x}$, which is applicable to first-order probabilistic models. We call it a 'zero skip-mention sequence' because it includes all mentions from a document and there are no (i.e., zero) other mentions between any two consecutive mentions in it. To identify coreferent mentions, we first need to label it using the labels $\{O, C\}$. The label C states that the current mention is coreferent with the previous one, whereas O states that the current mention is not coreferent with the previous one. Our models are learned over these labels so as to be able to infer new labels for unseen mention sequences. Observe that for the selected example, first-order models could detect just three coreferent mentions $\{\text{there}, \text{It}, \text{DIY Market}\}$. To solve the problem of identification of coreferent mentions at longer distances (e.g., OBI and there), we introduce further transformations.

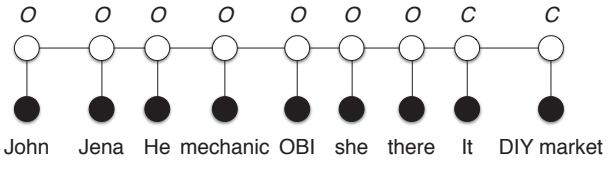

Fig. 8. **Zero skip-mention training sequence for coreference resolution.** Initial mention sequence that contains all mentions from the input text. If the current mention is coreferent with the previous one, it is labeled with *C*, otherwise with *O*.

We propose additional transformations, which generate multiple sequences and enable us to uncover all three clusters from the document x. Additional mention sequences are generated from the initial mention sequence $\overline{x}$ and are labeled accordingly, using $\{O, C\}$ labels. For instance, if we decide to use skip-mention distances ranging from zero to three, we transform the data set into four sequence types: zero, one, two and three skip-mention types. We also train a separate CRF model for each type, which enables us to tag new unseen data for specific skip-mention distance type. From the document x above, additional one skip-mention sequences are presented in Figure 9. By employing one skip-mention sequences, we extend our results by two new pairs $\{\texttt{John}, \texttt{He}\}$ and $\{\texttt{OBI}, \texttt{there}\}$. Then, after inference over two and three skip-mention sequences, we get our final missing pairs $\{\texttt{OBI}, \texttt{It}\}$ and $\{\texttt{Jena}, \texttt{she}\}$. Lastly, we perform mention clustering and return target entity clusters $x_{\texttt{John}}, x_{\texttt{Jena}}$ and $x_{\texttt{OBI}}$.

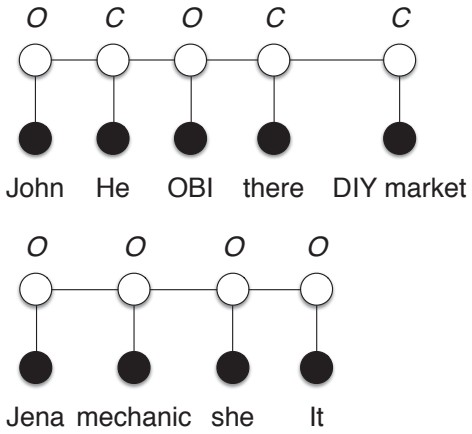

Fig. 9. **One skip-mention training sequences for coreference resolution.** Mention sequences that include every second mention (i.e., one skip-mention) from the input document. If the current mention is coreferent with the previous one, it is labeled with *C*, otherwise with *O*.

To support our idea, we show the distribution of distances between two consecutive coreferent mentions (see Figure 10) in the ACE 2004 data set. Although the figure shows the distribution for only the selected data set, it is representative enough to illustrate the general problem, which is the same for all other existing coreference resolution data sets (e.g., SemEval 2010, CoNLL 2012). According to the distribution, only 35% of the directly consecutive mention pairs are coreferent. Taking into account all mention pairs up to a distance of 20, cumulatively, 90% of the mention pairs can be identified. With distances up to 50, about 97% of the mention pairs can be identified. However, by using longer or all possible distances, the accuracy of a general coreference system is not expected to increase since there are more precision errors. To overcome such problems, we select a promising cut-off point at distance of 25 (see Figure 11).

As shown in the example above, the transformation into higher skip-mention sequences returns more sequences per document. Intuitively, at distance zero, we get one training sequence per document (it contains all document mentions). At distance one, we get two sequences (each contains every second mention). At distance two, we get three sequences, etc. Therefore, the transformation into d skip-mention sequences returns $d+1$ sequences of length $\lceil \frac{n}{d} \rceil$, where n is the number of all mentions in the document.

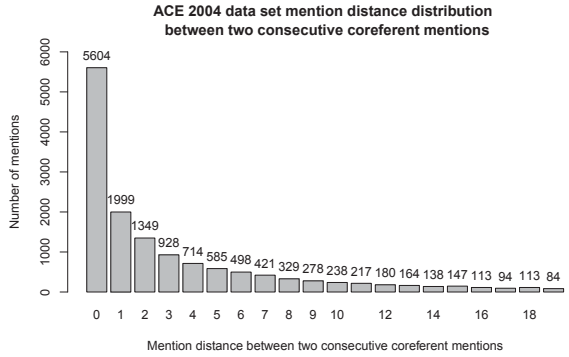

Fig. 10. **Distribution of distances between two consecutive coreferent mentions on ACE 2004 data set.** Distance x between two consecutive mentions means that there exist x other mentions between them. The consecutive coreferent mentions with 20 or more mention distances are truncated for better visualization (i.e., 1533 mentions).

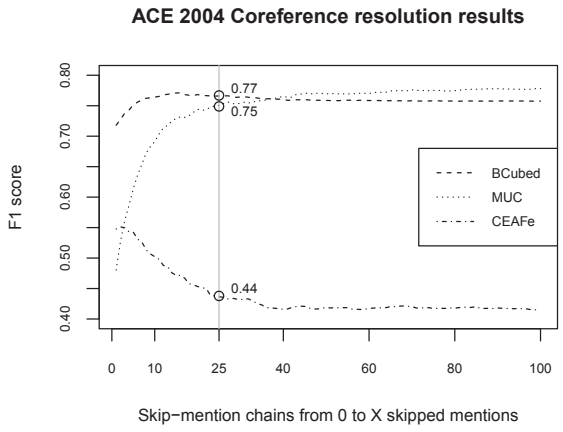

Fig. 11. **Coreference resolution results using different skip-mention sequences.** Evaluation of the proposed coreference resolution approach on the ACE 2004 dataset.

We use LCRF models and skip-mention sequences as input to the models for both relationship extraction and coreference resolution. Thus, the extraction technique is similar for both tasks. The main difference is in the target sequence labeling and representation of the tagged data. For relationship extraction task we check tagged labels to infer extracted relations between the input mentions. For the coreference resolution task we perform agglomerative clustering based on tagged labels to infer clusters of mentions, which represent the final entities. For a high level representation of data flow for both tasks using the proposed approach, see Figure 12.

B. Proposed system

We propose a modular end-to-end information extraction system that jointly and iteratively combines multiple extractors (Figure 13). The system consists of preprocessing component, three extractors that use an ontology via semantic feature functions, and data integration component. The input to the system consists of unstructured text documents and the system returns a graph-based representation of extractions that comply with the system ontology. Full implementation of the system is publicly available [54] and uses CRFSuite [55] for faster CRF training and inference and OpenNLP toolkit [56] for preprocessing tasks.

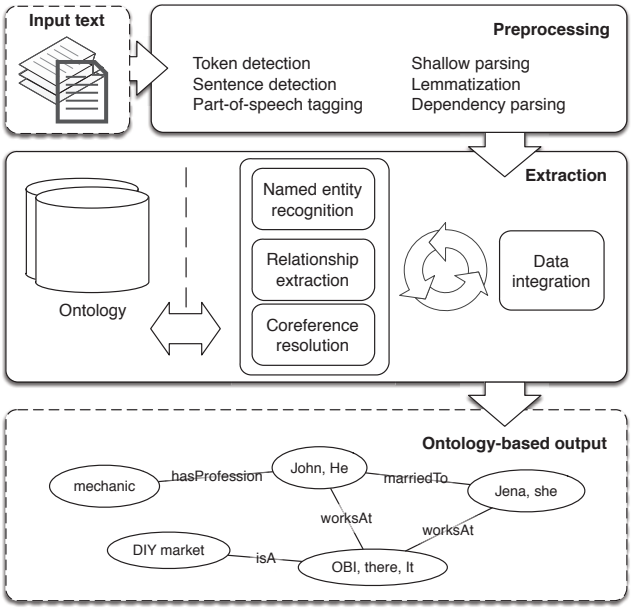

Fig. 13. **Iterative joint information extraction system.** The system consists of preprocessing and extraction components. The extraction component iteratively connects named entity recognition, relationship extraction and coreference resolution using a data integration method, and provides connection to the system ontology.

1) Preprocessing: 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, mention detection, and shallow parsing. 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 be able to identify sentence and word boundaries.

2) Extraction: The extraction consists of the iterative method and data integration method that combines intermediate extractions, which can be used for further extractions to improve the system performance of the system. In the iterative method we separately employ named entity recognition, relationship extraction, and coreference resolution extractors. The extraction is performed on the same data set and with the same type of LCRF classifier for all three tasks, so there is no need for special transformations between the tasks. We only need to map tokens with mentions because named entity recognizer works over token-level sequences, while relationship extractor and coreference resolution method work over mention-level sequences. When mapping from a token to a mention, we find a mention which contains the token an uses its attributes. Otherwise, when mapping from a mention to a token, we use attributes of the first token in the mention (i.e., mention is a sequence of one or more tokens). After each iteration, the data integration method performs tagging with classifiers that were built during the current iteration and updates iteration

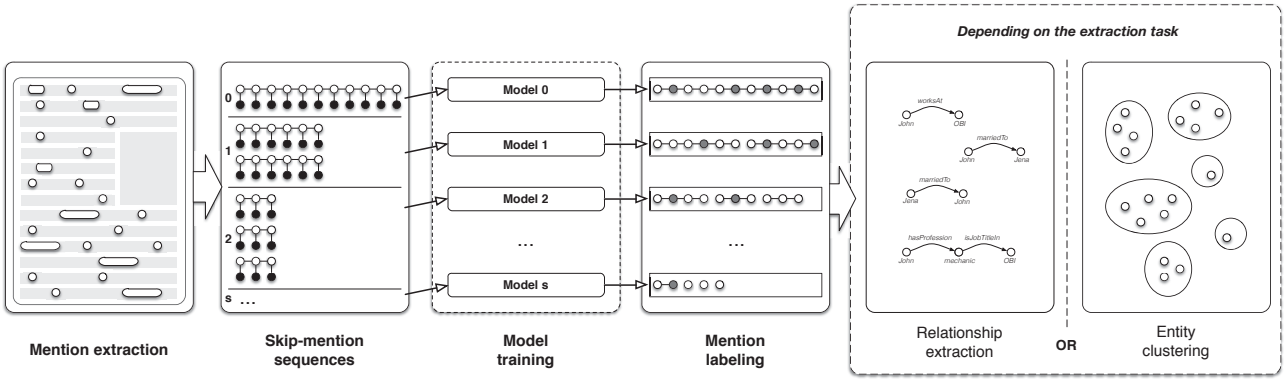

Fig. 12. **Data flow using LCRF-based approach for relationship extraction or coreference resolution.** First, the initial skip-mention sequence is transformed into the selected skip-mention sequences. Then, for each of the skip-mention sequence type, a different LCRF model is trained and further used to label the appropriate skip-mention sequences. The last step depends on the selected task for which the models were learned. For relationship extraction task, relationships are instantiated from the tagged sequences and returned as a result. For coreference resolution task, the mentions are clustered and each cluster of mentions represents a specific entity.

target label values for each of the subtasks. These intermediate labelings are then used by the iterative feature functions.

Ontology module is used in three different contexts. Firstly, the ontology represents the underlying domain modeled by the information extraction 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, parent concepts or gazetteer list of known instances) that are used directly by feature functions and thus system performance can be improved by the ontology manipulation. When the system is deployed in a production environment, this is the only part of it that can be manipulated by a user. Lastly, ontology also serves as a data store schema for extracted results.

The results of the extraction are iteratively added to the input data. At the end of the method execution the extractions are read from the data set and returned as a semantic graph, according to the system ontology.

3) Feature functions: Although many feature functions have been proposed in the literature [6], [29], [57], [58], [59], [60], we introduce new feature functions for the purpose of this research. These can be sorted into the following categories:

Preprocessing. These feature functions use standard preprocessing labels, which are a result of the preprocessing step, such as lemmas, part-of-speech (POS) tags, chunks, and parse trees. The derived feature function examples are "target label distribution", "do POS tags match on distances up to two mentions away", "distribution of POS tags", "mention type match", "is a mention pronoun of demonstrative/definitive noun phrase", "is mention a pronoun", "length between mentions within a parse tree", "parse tree path from the root node", "parse tree path between the two mentions", "depth of a mention within a parse tree", and "parse tree parent value match".

Location. Sometimes it is important to know where the mention resides. Location feature functions deal with the mention's location compared to the whole document, sentence, or other mentions. Our approach already implicitly uses mention distance at each skip-mention model, but we still employ some specific feature functions. Some of them are "sentence, mention or token distance between the two mentions", "is first/last mention" and "are mentions within the same sentence".

Mention shape. Mention constituents are represented as word phrases and by using mention shape features we are interested in whether two of them share some property. These feature functions are string-based and some of them are implemented as follows: "does a mention start with an upper case", "do both mentions start with upper case", "does a prefix, postfix, whole of left or right mention on distances up to five mentions match", "does a mention text or extent match", "is one mention appositive of another", "is one mention prefix, suffix or substring of another", "Hearst mention co-occurrence rules", "is a mention within quotes", "does a mention contain head or extent words of another" and "length difference between the two mentions".

Semantic. This class of feature functions captures semantic relationships between mentions by employing additional semantic sources, such as WordNet [61], specialized lexicons, semantic gazeteer lists, and the system ontology. Some semantic feature functions are "do named entity types match", "do mentions agree on gender/number" [62], "is one mention appositive of another", "is a mention an alias of another" (heuristically), "edit distance similarity between two mentions", "WordNet relation (hypernym, hyponym or synonym) between the mentions", "do mentions share the same WordNet synset", "current mention word sense", "do both mentions represent an animate object" [63], "do both mentions speak" (taking context words into account), "concept type from the ontology", "parent concept type from the ontology", and "possible existing relationships between the ontology instances".

Iterative. The iterative feature functions enable joint information extraction because all the intermediate extractions from a previous iteration are used in a current iteration. The iterative feature functions we use include, for example, "named entity tags", "extracted relationship between the two mentions", "extracted relationship name and observable

values", "labels of other mention in an extracted relationship", and "labels of mentions in the same coreferent cluster".

Due to the large number of different feature functions, a detailed description is omitted. Still, their exact implementations can be retrieved from our public source repository [54] (within the class `FeatureFunctionPackages`).

V. Results and discussion

To evaluate the proposed system we use Automatic Content Extraction (ACE) 2004 data set [64] because it is to our knowledge the only one that includes manually labeled named entities, relationships and coreferences. For the purposes of evaluation we used all four distinct news domains of English version of the data set. The selected data set consists of 443 documents with 191,387 tokens and 7,518 sentences. Further details regarding the used data set are shown in Table I. The named entities we extract are as follows: person, organization, facility, location, and geopolitical entity. From the relationships we extract five general relations: physical, personal/social, employment/membership/subsidiary, agent-artifact, person/organization affiliation, geopolitial entity affiliation, and discourse. In the evaluation we use gold mention boundaries and select 80% of data for training and 20% of data for testing.

In Tables II, III and IV we show extraction results after each of the five iterations. After the first iteration we achieve the same results as if we would run each extractor independently because no previous iteration exist. Extractors use feature functions that are described in the previous section, relationship extractor uses skip-mention sequences ranging from zero to three and coreference extractor ranging from zero to twenty-five.

TABLE II. Named entity recognition results.

Model	Error reduction (%)	CA	MaP	MaR	MaF	MiF
Independent	–	97.0	54.0	30.4	38.9	90.8
Second iteration	10.3	97.2	55.0	33.2	41.4	91.6
Third iteration	15.0	97.8	55.2	33.5	41.7	92.2
Fourth iteration	15.0	97.8	55.2	33.5	41.7	92.2
Fifth iteration	15.0	97.8	55.2	33.5	41.7	92.2

The shown metrics are error reduction in %, classification accuracy (CA), macro-averaged precision (MaP), macro-averaged recall (MaR), macro-averaged F score (MaF) and micro-averaged F score (MiF).

We report the results using metrics that are used by the most of researchers. For named entity recognition we use macro and micro-averaged F score because these metrics can show errors when classifying into different classes (Table II). Classification accuracy of 97.8% does not tell a lot about the result as there are many tokens in the text that do no belong to a specific named entity (t.i., tokens labeled with *O*). It is interesting to see, that up to the third iteration, the error count is reduced by 15%. Similar results are also reported by Singh et al. [47]. Macro-averaged score treats all named entity types across all the documents as equal, while micro-averaged score averages the results from all of the documents per entity type. This is also why there is a big difference between the two measures. Macro-average F score is low because some named entity types are poorly recognized. Although, the results nicely show that the performance of the system is improved after iterations.

TABLE III. Relationship extraction results.

Model	Error reduction (%)	P	R	F
Independent	–	54.3	55.2	54.7
Second iteration	0.8	55.1	55.6	55.3
Third iteration	2.4	54.8	55.6	55.2
Fourth iteration	2.4	55.0	55.4	55.2
Fifth iteration	2.0	54.2	55.7	54.9

The shown metrics are error reduction in %, precision (P), recall (R) and F score (F).

For the relationship extraction we use a standard F score metric, which is a harmonic mean between precision and recall. We will recognize a relationship as correctly extracted if it matches by type and by both attributes. Also for this information extraction task we improve the results by a little over the iterations (Table III). Also, the number of errors decreases but not as significantly as for named entity recognition task. At the named entity recognition we observe that the results do not increase after the third iteration, while at relationship extraction, the results are even decreased in the fifth iteration.

TABLE IV. Coreference resolution results.

Model	MUC	BCubed	CEAF
Independent	73.2	73.9	49.8
Second iteration	73.8	73.5	50.0
Third iteration	**74.0**	**74.1**	**52.9**
Fourth iteration	74.3	73.8	52.8
Fifth iteration	74.3	73.8	52.8

Coreference resolution systems are evaluated using the metrics MUC [65], BCubed [66] and CEAF [67].

For the coreference resolution we achieve 1.9% lower MUC score and 5.2% lower BCubed score comparing to existing best results reported by Haghighi and Klein [33]. Such results are expected because our approach does not use ground truth named entities and relationships from the dataset, but uses results from the other two methods that are partly incorrect. Nevertheless, also the results for this task also show similar trend of increasing results as for other two tasks. After the third iteration the BCubed and CEAF results decrease, while MUC score is increasing. That is why the MUC metric assigns better scores to bigger entities even though they are not clean.

We observe that after each iteration the results slightly improve for each of the information extraction task, which is our goal. We expect that it is reasonable to use only three iterations because in the second iteration the results from other tasks are taken into account and in the third iteration these results are propagated through two classifiers. For example, in the first iteration we recognize a named entity, which could enable an extraction of a previously unextracted relationship. Then in the third iteration we could better solve a coreference resolution problem as we know an additional relationship type for a mention. All the improvements we report are achieved only by a joint information extraction over more iterations. The most obvious results are seen for named entity recognition, where we achieve a reduction of error of 15%. We also investigated the differences in results if we disable one of the extractor so that no named entity was recognized, no relationship was extracted or each mention represented its own entity. As the differences between the iterations are already so small, we could not identify, which task is the most important

TABLE I. ACE 2004 DATA SET PROPERTIES.

Data	#Documents	#Sentences	#Tokens	#Mentions	#Entities	#Relationships
Train	354	5,789	140,237	22,079	9,544	4,619
Test	89	1,638	35,490	5,679	2,488	1,185

for the significant increase of results by other two tasks. In the further work we will try to additionaly improve the results over the iterations and then show the dependencies between the information extraction tasks.

VI. CONCLUSIONS

The present paper proposed a novel end-to-end information extraction system. The system combines the named entity recognition, relationship extraction and coreference resolution in a joint and iterative extraction method. We further introduced special transformations of data into skip-mention sequences to enable the training and inference using simple first-order graphical models for all three tasks. The system also includes an ontology that serves as a database of existing knowledge and as a schema for extracted results.

The proposed system was evaluated on the only existing standard data set ACE 2004, which contains manually labeled data for all three tasks. Our implementations of each extractor already achieved comparable results to existing work individually and their performance was then improved by about 1% after three iterations of joint extraction. Furthermore, we noticed a high level of error reduction of about 15% for named entity recognition task.

The future work will focus on the improvement of the extractors in order to make them even more interconnected. Moreover, we will also work on developing a scoring measure that could serve as a standard model for evaluating end-to-end information extraction systems.

ACKNOWLEDGEMENTS

The work has been supported by the Slovene Research Agency (P2-0359).

REFERENCES

[1] P. M. Andersen, P. J. Hayes, A. K. Huettner, L. M. Schmandt, I. B. Nirenburg, and S. P. Weinstein, "Automatic extraction of facts from press releases to generate news stories," in *Proceedings of the third conference on Applied natural language processing*. Pennsylvania: Association for Computational Linguistics, 1992, pp. 170–177.

[2] S. Sarawagi, "Information extraction," *Foundations and Trends in Databases*, vol. 1, no. 3, pp. 261–377, 2008.

[3] D. C. Wimalasuriya and D. Dou, "Ontology-based information extraction: An introduction and a survey of current approaches," *Journal of Information Science*, vol. 36, no. 3, pp. 306–323, 2010.

[4] C. Ndellec, R. Bossy, J.-D. Kim, J.-J. Kim, T. Ohta, S. Pyysalo, and P. Zweigenbaum, "Overview of BioNLP Shared Task 2013," in *Proceedings of BioNLP Shared Task 2013 Workshop*, 2013, pp. 1–7.

[5] J. Cai and M. Strube, "End-to-end coreference resolution via hypergraph partitioning," in *Proceedings of the 23rd International Conference on Computational Linguistics*. Pennsylvania: Association for Computational Linguistics, 2010, pp. 143–151.

[6] V. Ng, "Unsupervised models for coreference resolution," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Pennsylvania: Association for Computational Linguistics, 2008, pp. 640–649.

[7] X. Luo, "Coreference or not: A twin model for coreference resolution," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*. New York: Association for Computational Linguistics, 2007, pp. 73–80.

[8] 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 Fifteenth Conference on Computational Natural Language Learning: Shared Task*. Pennsylvania: Association for Computational Linguistics, 2011, pp. 28–34.

[9] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel, "The automatic content extraction (ACE) programtasks, data, and evaluation," in *Proceedings of LREC*, vol. 4. Paris: European Language Resources Association, 2004, p. 837840.

[10] M. Recasens, L. Mrquez, E. Sapena, A. M. Mart, M. Taul, V. Hoste, M. Poesio, and Y. Versley, "SemEval-2010 task 1: Coreference resolution in multiple languages," in *Proceedings of the 5th International Workshop on Semantic Evaluation*. Stroudsburg: Association for Computational Linguistics, 2010, pp. 1–8.

[11] S. Pradhan, A. Moschitti, N. Xue, O. Uryupina, and Y. Zhang, "CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes," in *Proceedings of the Joint Conference on EMNLP and CoNLL: Shared Task*. Stroudsburg: Association for Computational Linguistics, 2012, pp. 1–40.

[12] L. Smith, L. K. Tanabe, R. Ando, C.-J. Kuo, I.-F. Chung, C.-N. Hsu, Y.-S. Lin, R. Klinger, C. M. Friedrich, K. Ganchev, M. Torii, H. Liu, B. Haddow, C. A. Struble, R. J. Povinelli, A. Vlachos, W. A. Baumgartner, L. Hunter, B. Carpenter, R. Tsai, H.-J. Dai, F. Liu, Y. Chen, C. Sun, S. Katrenko, P. Adriaans, C. Blaschke, R. Torres, M. Neves, P. Nakov, A. Divoli, M. Maa-Lpez, J. Mata, and W. J. Wilbur, "Overview of BioCreative II gene mention recognition," *Genome Biology*, vol. 9, no. Suppl 2, p. S2, 2008.

[13] C. Ndellec, "Learning language in logic-genic interaction extraction challenge," in *Proceedings of the 4th Learning Language in Logic Workshop (LLL05)*, vol. 7. Bonn: ACM, 2005, pp. 1–7.

[14] B. Liu, L. Chiticariu, V. Chu, H. V. Jagadish, and F. R. Reiss, "Automatic rule refinement for information extraction," *Proceedings of the VLDB Endowment*, vol. 3, no. 1, 2010.

[15] N. Dalvi, R. Kumar, and M. Soliman, "Automatic wrappers for large scale web extraction," *Proceedings of the VLDB Endowment*, vol. 4, no. 4, p. 219230, 2011.

[16] M. Vazquez, M. Krallinger, F. Leitner, and A. Valencia, "Text mining for drugs and chemical compounds: Methods, tools and applications," *Molecular Informatics*, vol. 30, no. 6-7, pp. 506–519, 2011.

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

[18] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Pennsylvania: Association for Computational Linguistics, 2005, p. 363370.

[19] E. Fosler-Lussier, Y. He, P. Jyothi, and R. Prabhavalkar, "Conditional random fields in speech, audio, and language processing," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1054–1075, 2013.

[20] J. Jiang and C. Zhai, "A systematic exploration of the feature space for relation extraction." in *HLT-NAACL*, 2007, p. 113120.

[21] M. Garcia and P. Gamallo, "Dependency-based text compression for semantic relation extraction," *Information Extraction and Knowledge Acquisition*, p. 21, 2011.

[22] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction," *The Journal of Machine Learning Research*, vol. 3, p. 10831106, 2003.

[23] R. C. Bunescu and R. J. Mooney, "A shortest path dependency kernel

for relation extraction," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 2005, p. 724731.

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

[25] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the web," *Procs. of IJCAI*, 2007.

[26] M. Banko and O. Etzioni, "The tradeoffs between open and traditional relation extraction," *Proceedings of ACL-08: HLT*, p. 2836, 2008.

[27] A. Culotta, M. Wick, R. Hall, and A. McCallum, "First-order probabilistic models for coreference resolution," in *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 2007, pp. 81–88.

[28] V. Ng and C. Cardie, "Improving machine learning approaches to coreference resolution," in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Philadelphia: Association for Computational Linguistics, 2002, pp. 104–111.

[29] E. Bengtson and D. Roth, "Understanding the value of features for coreference resolution," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Pennsylvania: Association for Computational Linguistics, 2008, pp. 294–303.

[30] X. Luo, A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos, "A mention-synchronous coreference resolution algorithm based on the bell tree," in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Pennsylvania: Association for Computational Linguistics, 2004, pp. 136–143.

[31] A. Rahman and V. Ng, "Supervised models for coreference resolution," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, vol. 2, 2009, pp. 968–977.

[32] S. Huang, Y. Zhang, J. Zhou, and J. Chen, "Coreference resolution using markov logic networks," *Advances in Computational Linguistics*, vol. 41, pp. 157–168, 2009.

[33] A. Haghighi and D. Klein, "Simple coreference resolution with rich syntactic and semantic features," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, vol. 3. Pennsylvania: Association for Computational Linguistics, 2009, pp. 1152–1161.

[34] A. McCallum and B. Wellner, "Conditional models of identity uncertainty with application to noun coreference," in *Neural Information Processing Systems*, 2004, pp. 1–8.

[35] B. Wellner, A. McCallum, F. Peng, and M. Hay, "An integrated, conditional model of information extraction and coreference with application to citation matching," in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. Arlington: AUAI Press, 2004, pp. 593–601.

[36] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by Gibbs sampling," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Pennsylvania: Association for Computational Linguistics, 2005, pp. 363–370.

[37] P. Cimiano, U. Reyle, and J. ari, "Ontology-driven discourse analysis for information extraction," *Data & Knowledge Engineering*, vol. 55, no. 1, pp. 59 – 83, 2005, natural Language and Database and Information Systems {NLDB} 03.

[38] L. McDowell and M. Cafarella, "Ontology-driven information extraction with ontosyphon," *The Semantic Web-ISWC 2006*, p. 428444, 2006.

[39] A. Ittoo and G. Bouma, "Minimally-supervised extraction of domain-specific partwhole relations using wikipedia as knowledge-base," *Data & Knowledge Engineering*, vol. 85, no. 0, pp. 57 – 79, 2013, natural Language for Information Systems: Communicating with Anything, Anywhere in Natural Language.

[40] B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, and A. Kirilov, "Kim – a semantic platform for information extraction and retrieval," *Nat. Lang. Eng.*, vol. 10, no. 3-4, pp. 375–392, 2004.

[41] S. Saha and A. Ekbal, "Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition," *Data & Knowledge Engineering*, vol. 85, no. 0, pp. 15 – 39, 2013, natural Language for Information Systems: Communicating with Anything, Anywhere in Natural Language.

[42] 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*, ser. ACL '04. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004.

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

[44] X. Yu and W. Lam, "Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, 2010.

[45] D. Roth and W.-t. Yih, "Global inference for entity and relation identification via a linear programming formulation," *Introduction to statistical relational learning*, p. 553580, 2007.

[46] L. Yao, S. Riedel, and A. McCallum, "Collective cross-document relation extraction without labelled data," 2010.

[47] S. Singh, S. Riedel, B. Martin, J. Zheng, and A. McCallum, "Joint inference of entities, relations, and coreference." ACM Press, 2013, pp. 1–6.

[48] C. Sutton and A. McCallum, "Joint parsing and semantic role labeling," 2005.

[49] H. Poon and P. Domingos, "Joint inference in information extraction," in *AAAI*, vol. 7, 2007, p. 913918.

[50] S. Singh, K. Schultz, and A. McCallum, "Bi-directional joint inference for entity resolution and segmentation using imperatively-defined factor graphs," in *Machine Learning and Knowledge Discovery in Databases*, 2009, p. 414429.

[51] H. Poon and L. Vanderwende, "Joint inference for knowledge extraction from biomedical literature," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010, p. 813821.

[52] S. Riedel and A. McCallum, "Fast and robust joint models for biomedical event extraction," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, p. 112.

[53] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*. San Francisco: Morgan Kaufmann, 2001, pp. 282–289.

[54] S. Žitnik, "Intelligent Ontology-based Information Extraction - IOBIE, source code repository," 2013, available from https://bitbucket.org/szitnik/iobie (last accessed 17th April 2014).

[55] N. Okazaki, "CRFsuite: a fast implementation of conditional random fields (CRFs)," 2007, available from http://www.chokkan.org/software/crfsuite (last accessed 17th April 2014).

[56] "Apache OpenNLP: a machine learning based toolkit for the processing of natural language text," available from http://opennlp.apache.org/ (last accessed 17th April 2014).

[57] W. M. Soon, H. T. Ng, and D. C. Y. Lim, "A machine learning approach to coreference resolution of noun phrases," *Computational linguistics*, vol. 27, no. 4, pp. 521–544, 2001.

[58] S. Broscheit, M. Poesio, S. P. Ponzetto, K. J. Rodriguez, L. Romano, O. Uryupina, Y. Versley, and R. Zanoli, "BART: a multilingual anaphora resolution system," in *Proceedings of the 5th International Workshop on Semantic Evaluation*. Pennsylvania: Association for Computational Linguistics, 2010, pp. 104–107.

[59] G. Attardi, S. D. Rossi, and M. Simi, "TANL-1: coreference resolution by parse analysis and similarity clustering," in *Proceedings of the 5th International Workshop on Semantic Evaluation*. Pennsylvania: Association for Computational Linguistics, 2010, pp. 108–111.

[60] E. R. Fernandes, C. N. dos Santos, and R. L. Milidiú, "Latent structure perceptron with feature induction for unrestricted coreference resolution," in *Proceedings of CoNLL 2012 Joint Conference on EMNLP and CoNLL*. Pennsylvania: Association for Computational Linguistics, 2012, pp. 41–48.

[61] G. A. Miller, "WordNet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[62] S. Bergsma and D. Lin, "Bootstrapping path-based pronoun resolution," in *Proceedings of the 21st International Conference on Computational*

Linguistics and the 44th annual meeting of the Association for Computational Linguistics. Pennsylvania: Association for Computational Linguistics, 2006, pp. 33–40.

[63] C. Orasan and R. Evans, "NP animacy identification for anaphora resolution," *Journal of Artificial Intelligence Research*, vol. 29, no. 1, pp. 79–103, 2007.

[64] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel, "The automatic content extraction (ACE) programtasks, data, and evaluation," in *Proceedings of LREC*, vol. 4. Paris: European Language Resources Association, 2004, p. 837840.

[65] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman, "A model-theoretic coreference scoring scheme," in *Proceedings of the 6th conference on Message understanding*. Pennsylvania: Association for Computational Linguistics, 1995, pp. 45–52.

[66] A. Bagga and B. Baldwin, "Algorithms for scoring coreference chains," in *The first international conference on language resources and evaluation workshop on linguistics coreference*, vol. 1, 1998, pp. 1–7.

[67] X. Luo, "On coreference resolution performance metrics," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Pennsylvania: Association for Computational Linguistics, 2005, pp. 25–32.