J-REED: Joint Relation Extraction and Entity Disambiguation

Dat Ba Nguyen
MPI for Informatics
datnb@mpi-inf.mpg.de

Martin Theobald
University of Luxembourg
martin.theobald@uni.lu

Gerhard Weikum
MPI for Informatics
weikum@mpi-inf.mpg.de

ABSTRACT

Information extraction (IE) from text sources can either be performed as *Model-based IE* (i.e, by using a pre-specified domain of target entities and relations) or as *Open IE* (i.e., with no particular assumptions about the target domain). While Model-based IE has limited coverage, Open IE merely yields triples of surface phrases which are usually not disambiguated into a canonical set of entities and relations. This paper presents *J*-REED: a *joint* approach for entity disambiguation and relation extraction that is based on *probabilistic graphical models*. *J*-REED merges ideas from both Model-based and Open IE by mapping surface names to a background knowledge base, and by making surface relations as crisp as possible.

CCS CONCEPTS

• **Information systems** → *Content analysis and feature selection*;

KEYWORDS

Open relation extraction, entity disambiguation, joint inference

1 INTRODUCTION & RELATED WORK

Motivation. Information extraction (IE) aims to distill relational triples, each consisting of an entity pair (or an entity and a literal value) plus a connecting relation, from natural-language text. This goal has been pursued by two major approaches. *Model-based IE* [5, 7, 8, 18, 23, 25] focuses on a set of pre-specified relations, like those present in a knowledge base (KB) such as DBpedia [1], NELL [7] or Yago [24]. Entity names recognized in the input text are disambiguated by mapping them to proper entities in the KB. The relations have fine-grained type signatures that need to be matched by the assigned entities. Model-based IE techniques leverage this strategy to achieve high precision, but they are inherently limited in recall by the relatively small amount of given relations. *Open IE* [2, 10, 17], on the other hand, extracts triples of surface phrases and thus achieves higher recall. It can potentially find any relation that holds between two arguments. However, the arguments are not canonicalized, and the resulting relations are often noisy.

Example. Consider the following four input sentences:
(1) *Amy received the Oscar for the best documentary.*
(2) *Amy received the Grammy for the best new artist.*
(3) *Amy received her degree in neurobiology from Harvard.*
(4) *Simone received honorary degrees in music and humanities, from both UMass Amherst and Malcolm X College.*

Sentences (1), (2) and (3) refer to different entities—the movie, the singer and the movie character—, which are all named "*Amy*". The sentences provide cues for identifying the lexical types, but no existing IE method can robustly handle such cases. Moreover, (3) and (4) express the same relation (i.e., "*receive degree from*") by different phrases, but Open IE would treat them as separate relations.

State-of-the-Art & Limitations. Recent work aimed to reconcile Model-based IE and Open IE. Hoffmann et al. [15] present a distantly supervised IE system which can handle thousands of relations by clustering the relational paraphrases based on their arguments and type signatures (e.g., using 1,282 such clusters in an experiment). However, this approach is highly customized to using Wikipedia infoboxes as input. Galárraga et al. [13] canonicalize Open IE triples by clustering noun phrases into arguments and verbal phrases into relations. This approach is limited, though, to a few hundred relation clusters by using Freebase [4] as backend for the relations. Li and Ji [16], on the other hand, jointly extract entity and relation names, but refrain from disambiguating these.

None of the prior works considers *joint inference* to extract relations and to disambiguate the entities in one step. Instead, all of the prior works use pipelined architectures and, thus, cannot fully harness the coupling of lexical types for entities with the type signatures for relations. For example, to properly distinguish the entities *Amy_Winehouse* and *Amy_Farrah_Fowler* in the above examples (2) and (3), understanding the different type signatures of the relations "*receive prize*" (`SINGER` × `MUSIC AWARD`) and "*receive degree from*" (`PERSON` × `UNIVERSITY`) is crucial. The approach proposed in this paper can leverage these kinds of interdependencies.

Contributions. This paper presents *J*-REED, a joint model for entity disambiguation and relation extraction for Wikipedia-style input texts. *J*-REED is based on *probabilistic graphical models* that captures the interdependencies between entities and relations. Specifically, by considering which lexical types of entities are compatible with the type signature of which relation, we can boost the accuracy of both sub-tasks. Entity names are mapped to the entities registered in a background KB (using DBpedia in our experiments), while relation patterns are extracted as crisp as possible. We performed large-scale experiments with 1.2 million Wikipedia pages about entities of type `PERSON` and obtained 9.5 million triples with ca. 80% accuracy. *J*-REED consistently outperforms pipelined combinations of OLLIE [17], a state-of-the-art Open IE system, and recent NED systems such as Babelfy [19] and Spotlight [9].

2 DOCUMENT PROCESSING

J-REED processes a text corpus in several steps. We first pass all documents through a standard NLP pipeline, including tokenization, POS tagging, dependency parsing, NER tagging, and a customized

CIKM'17 , November 6–10, 2017, Singapore, Singapore

ACM ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/10.1145/3132847.3133090

noun-phrase chunker. Specifically, we employ the Stanford CoreNLP tool suite for all of our text processing steps except for dependency parsing. For the latter, we use the MaltParser [22] which is more efficient than the Stanford parser. Mention names in the text are primarily recognized by the Stanford NER [12] tagger.

In addition, we implement a custom noun-phrase chunker (using a set of regular expressions over the POS tags) to also extract noun-phrases that do not overlap with any names extracted by the NER tagger. We remove noise among the obtained noun-phrases by keeping only those phrases that contain at least one *informative noun*, which we define to consist of the top 5% most frequent nouns in the current document that is processed. For example, within the Wikipedia article of `Amy_Winehouse`, the most informative nouns contain "*album*", "*alcohol*", etc. Thus, the noun phrase "*alcohol poisoning*" (i.e., the cause of her death) is considered as a mention even if the NER tagger missed this phrase.

By applying these preprocessing steps to a dedicated *development corpus* (which is disjoint from the collection of test documents used in our experiments), we also compute various (co-)occurrence statistics among nouns, verbs, prepositions and entities. These statistics are later used to mine the *relation patterns* (Section 3), and they further serve as input to the feature functions used for *relation pattern labeling* (Section 4), and *joint relation extraction and entity disambiguation* (Section 5).

3 RELATION PATTERN MINING

J-REED considers four types of relation patterns: verb (e.g., *marry*), verb-noun (e.g., *win prize*), verb-preposition (e.g. *play for*) and verb-noun-preposition (e.g., *win prize for*). Nouns, prepositions and verbs in *active voice* are considered in their lemmatized forms (e.g., *marry*). Verbs in *passive voice* are represented by their past participle (e.g., *married to*) to capture the inverse direction of the relationship. *J*-REED considers only *frequent relation patterns* occurring at least τ times in the development corpus. In our experiments, we set $\tau = 100$ and obtained 9,248 frequent patterns (out of 320,143 distinct ones). By only considering those patterns as relational candidates for the graphical models in the next sections, *J*-REED can extract concise relation patterns. For example, *J*-REED extracts the relation pattern "*receive degree from*" instead of the surface phrase "*received honorary degrees in music and humanities from*" which is considered by many other Open IE methods. This is useful for further applications such as KB construction, question answering, and others.

4 RELATION PATTERN LABELING

To extract a relation pattern from a sentence in the test corpus, we consider a dependency path between two mentions as a *sequence of tokens* $\mathbf{p} = \langle tok_1, \ldots, tok_n \rangle$. Thus, extracting relations from a sentence may be seen as a sequence labeling task in which we aim to find the best *sequence of labels* $\mathbf{l} = \langle l_1, \ldots, l_n \rangle$ by using four labels: `N` for a chosen noun, `V` for a chosen verb, `P` for a chosen preposition and `O` for "other". Each relation pattern must contain one `V` label with at most one `N` label and with at most one `P` label. To improve recall, *J*-REED considers two further heuristics.

- If only one `N` label is returned under the text pattern *NAME-LEFT's noun NAME-RIGHT* (e.g. "*Mary's son Bill Gates*"), we add "*have*" with label `V` to the beginning of the sequence (e.g., "*have son*").
- If no `V` label is returned after an apposition, we add "*be*" with label `V` to the beginning of the sequence (e.g., "*be daughter of*").

A linear-chain CRF model is built to obtain the labels, which we can then map to the relation patterns mined from the development corpus. The feature set consists of: tokens, POS tags, upper/lower case, being the first verb, being the last preposition, being an English word, following the preposition "*to*", following a verb, following any token with tag "*NNP*", and the previous label. These features have been widely used in prior work on Open IE [2, 17]. The CRF is trained by maximizing a conditional likelihood function [26] via L-BFGS optimization. Exact inference is feasible by a variant of the Viterbi dynamic-programming algorithm [3]. We remark that if we only considered the most likely relation candidate, our method would resemble a traditional Open IE task. However, in our joint model described in the next section, all relation pattern candidates (together with their weights) are considered as features.

5 JOINT MODEL

For the joint relation pattern extraction and entity disambiguation, *J*-REED considers an input triple consisting of a *pair of mentions* $\mathbf{m} = \langle m_1, m_2 \rangle$ together with the *sequence of tokens* $\mathbf{p} = \langle tok_1, \ldots, tok_n \rangle$ along the dependency path that connects these mentions. The output of *J*-REED consists of a *pair of entities* $e_1 \in \mathbf{e_1}$, $e_2 \in \mathbf{e_2}$ and a *relation* $r \in \mathbf{r}$ that represents the semantic relationship between e_1 and e_2. Relations in $\mathbf{r}$ are obtained from the output of the CRF model described in Section 4, while entity candidates in $\mathbf{e_1}$ and $\mathbf{e_2}$ are obtained from the name-entity dictionary which is based on surface forms and hyperlinks in Wikipedia. To improve recall, we enrich this dictionary by including also the first and last names of all person entities in Wikipedia.

Model. *J*-REED considers five features: one feature for relation extraction (namely, *relation prior*), three features for entity disambiguation (namely, *mention-entity prior*, *mention-entity token context similarity* and *entity-entity token context similarity*) and one feature for the joint inference (namely, *type signature*). We remark that some recent NED features such as domain-oriented features [20] and syntactic dependency features [21] are not considered by *J*-REED as the former is less effective in Wikipedia-style texts, and the latter is embedded in type signature. The *type signature* feature captures type dependencies between entities (i.e., arguments) and relation patterns among those entities. This feature is used in some prior joint models of NED and Information Extraction which focus on a small set of pre-defined relations. Here, we further extend this technique by harnessing the type signatures for thousands of relations. Specifically, we define the *type signature* as the relative frequency at which the semantic types of two entities occur under a relation pattern. We obtain these frequency statistics from the development corpus. In addition to the four general NER types `PERSON`, `ORGANIZATION`, `LOCATION`, and `MISC`, we use frequent infobox types[1] that have at least 1,000 articles in Wikipedia. From the resulting 167 types, we also manually derive a subsumption hierarchy (e.g. `FOOTBALLER` $\subseteq$ `SPORTS-PERSON` $\subseteq$ `PERSON`).

[1] https://en.wikipedia.org/wiki/Wikipedia:List_of_infoboxes

These features are combined into the following *objective function*

$$\langle e_1, e_2, r\rangle^* = \underset{\langle e_1, e_2, r\rangle}{\arg\max} \sum_{i=1}^{5} \alpha_i h_i(\mathbf{m}, \mathbf{p}, e_1, e_2, r) \quad (1)$$

where the h_i are the features described above and the α_i are the parameters of our joint model.

Training & Inference. The α_i parameters are learned by maximizing the probability of the ground-truth annotations under L-BFGS optimization. At extraction time, exact inference is performed by dynamic programming [3].

6 EXPERIMENTS

6.1 Corpora

J-REED is applicable to any corpus with entity-centric documents, such as homepages of people, companies, etc. In our experiments we focused on Wikipedia articles about people. We remark that this does not restrict *J*-REED to `PERSON` entities. *J*-REED extracts various types of entities (e.g., `PERSON`, `MOVIE`, `AWARD`, etc.), and various types of relations (e.g., "*receive award*": `PERSON x AWARD`).

Development & Training Corpus. For the various stages of training, tuning and gathering statistics, we considered a collection of 1,215,956 Wikipedia articles about `PERSON` entities (based on types in Yago [14]) from the 01/2015 English Wikipedia dump. 80% of these articles with 19,287,432 triples from 8,312,439 sentences, called "*Wikipedia-develop* ", were used to develop *J*-REED (i.e., to compute various (co-)occurrence statistics for nouns, verbs, prepositions and entities). In addition, we manually annotated 162 sentences from 5 of these Wikipedia articles, referred to as "*J-REED-train*", about prominent person entities, namely, Andrew Ng, Angela Merkel, David Beckham, Larry Page and Paris Hilton (covering scientists, politicians, sports stars, business people and actors/singers/celebrities). These annotations comprise 203 triples, each consisting of two DBpedia entities and a sequence of `N`, `V`, `P`, `O` labels along the dependency path that connects the two entities. *J-REED-train* serves to train the CRF model (Section 4) and the joint model (Section 5).

Test Corpus. The remaining 20% of the 1,215,956 articles, called "*Wikipedia-test* ", were used for evaluating our methods. We extracted 5,964,464 triples from 2,226,433 sentences from these articles.

Assessment. We had two judges who assessed our experimental results independently. They were shown sampled facts (i.e., two entities and a relation) from the output of different methods, along with the source sentence from which the fact was extracted. The judges were asked to label the fact as *true* only if all three components were correct, otherwise the fact was labeled as *false*. We observed strong inter-judge agreement. Cohen's kappa [6] was 0.7.

6.2 Systems under Comparison

We ran experiments with several *J*-REED variants to compare against the state-of-the-art Open IE approach (i.e., OLLIE [17]), and named entity disambiguation (NED) tools (i.e., Babelfy [19] and Spotlight [9]). We chose Spotlight and Babelfy due to their focus on DBpedia entities and their ability to process the output of an Open IE system. Other NED tools, such as TagMe [11], only work on plain text as input. Specifically, we compared the following *end-to-end* IE methods:

- *J*-REED is the joint model described in Section 5. We heuristically resolve pronouns (i.e., "*he*" and "*she*") by assuming that these always refer to the main entity of the article. This assumption is based on the common Wikipedia writing style (and carries over to other kinds of entity-centric documents such as people's homepages).
- *J*-REED-pipeline considers only the most likely relation label based on the CRF model described in Section 4. In other words, the relation is fixed before the entities are disambiguated. Pronoun resolution is considered as described above.
- OLLIE-Spotlight is a pipelined approach combining OLLIE and Spotlight.
- OLLIE-Babelfy is a pipelined approach combining OLLIE and Babelfy.

We also perform an ablation test with three settings:

- *J*-REED-nopronoun performs joint entity-relation disambiguation but omits pronoun resolution.
- *J*-REED-notype omits the *type signature* feature.
- *J*-REED-noprior omits the relation *prior* feature.

6.3 Experiments on Relation Pattern Extraction

We evaluated the *precision* on 100 randomly sampled results for Wikipedia-test . Precision is reported as the mean of a Wald interval at 95% confidence level. Moreover, we measured *recall* as the absolute number of relational triples extracted. This value implicitly reflects relative recall which is generally hard to estimate for a large-scale experiment. For both precision and recall, we considered only extractions that consisted of up to 6 tokens. Other extractions are usually noise. As shown in Table 1, *J*-REED outperforms both baselines in terms of both quality and number of extractions. *J*-REED-pipeline loses 4% in precision compared to *J*-REED . Many mistakes of OLLIE originate from ignoring type constraints of relations. We remark that, although the authors of OLLIE address this problem in their work [17], this feature is apparently turned off in their prototype software to increase recall.

Table 1: Experiments on Relation Pattern Extraction (confidence at 95%).

Method	Precision	#Extracts
J-REED	0.90 ± 0.05	1,931,462
J-REED-pipeline	0.86 ± 0.06	1,931,462
OLLIE	0.80 ± 0.07	1,646,231

6.4 Experiments on Entity Disambiguation

In a similar manner as for the relation pattern extraction, we also evaluated precision and recall of the entity disambiguation component. For this, we considered 1,931,462 triples as output from *J*-REED (see Subsection 6.3). As the other NED systems – Spotlight and Babelfy – do not map pronouns to entities, we ignored extractions containing pronouns. An extraction is considered to be correct only if both entities are disambiguated correctly. Table 2 shows that *J*-REED outperforms other methods. The overall difference in precision among the systems is not that large. Particularly, all systems achieve precision values of more than 80%. The numbers of disambiguated entity pairs are different for the systems, since we do not consider *null* entities in the final results.

Table 2: Experiments on Entity Disambiguation (confidence at 95%).

Method	Precision	#Extracts
J-REED	0.85 ± 0.06	1,931,462
J-REED-pipeline	0.82 ± 0.07	1,931,462
Spotlight	0.83 ± 0.07	1,036,319
Babelfy	0.81 ± 0.07	854,159

6.5 End-to-End Experiments

For this setting, we evaluated the correctness of entire facts (i.e., relational triples). In analogy to the above experiments, a fact extraction was considered correct only if all of its three components (i.e., the relation and the two entities) were correctly disambiguated. Here, we distinguished three types of assessments:

- *True* for a correct result,
- *False* for a wrong result (either the entities or the relation was wrong),
- *Ignored* for OLLIE-Spotlight and OLLIE-Babelfy when OLLIE returns a relation that consists of more than 6 tokens (i.e., usually noise).

Ignored results are not considered for precision and recall.

Ablation Test. As shown in Table 3, the *type signature* and *prior* features are crucial for the precision of *J*-REED. Disabling them results in a drop of precision by 6%. Disabling pronoun resolution penalizes the coverage by 40%. Additionally, we conduct experiments on the two heuristics for resolving possessive forms and appositions from Section 4. Disabling each of them penalizes the coverage by 6% and 12%, respectively.

Table 3: Ablation Test on Fact Extraction (confidence at 95%).

Method	Precision	#Extracts
J-REED	0.78 ± 0.08	1,931,462
J-REED-notype	0.72 ± 0.08	1,931,462
J-REED-noprior	0.75 ± 0.08	1,931,462
J-REED-nopronoun	0.80 ± 0.07	1,237,352

Comparison to Baselines. The *J*-REED variants clearly outperform OLLIE-Spotlight and OLLIE-Babelfy (Table 4). *J*-REED achieves around 5% higher precision than *J*-REED-pipeline . *J*-REED processes all 1.2 million Wikipedia pages in about five hours. Our competitors require more than a day to process the same data.

7 CONCLUSIONS

This paper presented *J*-REED, a large-scale high-quality IE system for Wikipedia-style text. Its unique strength is that it jointly runs inference for two core IE tasks: relation extraction and named-entity disambiguation by leveraging semantic types. Extractions by *J*-REED are more informative than by Open IE, as we canonicalize entities to a background knowledge base. Running *J*-REED on 1,215,956 Wikipedia `PERSON` pages yields 9,577,301 facts with a precision of around 80%.

Table 4: Experiments on Fact Extraction (confidence at 95%).

Method	Precision	#Extracts
J-REED	0.78 ± 0.08	1,931,462
J-REED-pipeline	0.73 ± 0.08	1,931,462
OLLIE-Spotlight	0.68 ± 0.09	690,409
OLLIE-Babelfy	0.67 ± 0.09	547,031

REFERENCES

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, and Zachary Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *Proceedings of the 6th Int'l Semantic Web Conference, Busan, Korea*. Springer, 11–15.

[2] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open Information Extraction from the Web. In *IJCAI '07*. Morgan Kaufmann Publishers Inc., 2670–2676.

[3] Richard Ernest Bellman. 2003. *Dynamic Programming*. Dover Publications Inc.

[4] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *SIGMOD '08*. ACM, 1247–1250.

[5] Sergey Brin. 1998. Extracting Patterns and Relations from the World Wide Web. In *The World Wide Web and Databases, International Workshop WebDB '98*. 172–183.

[6] Jean Carletta. 1996. Assessing Agreement on Classification Tasks: The Kappa Statistic. *Comput. Linguist.* 22, 2 (1996), 249–254.

[7] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. 2010. Toward an Architecture for Never-Ending Language Learning. In *AAAI '10*. AAAI Press.

[8] Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom M. Mitchell, Kamal Nigam, and Seán Slattery. 2000. Learning to construct knowledge bases from the World Wide Web. *Artif. Intell.* 118, 1-2 (2000), 69–113.

[9] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. 2013. Improving Efficiency and Accuracy in Multilingual Entity Extraction. In *Proceedings of the 9th International Conference on Semantic Systems, I-SEMANTICS '13*. ACM, 121–124.

[10] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. 2011. Open Information Extraction: The Second Generation. In *IJCAI '11*. AAAI Press, 3–10.

[11] Paolo Ferragina and Ugo Scaiella. 2010. TAGME: On-the-fly Annotation of Short Text Fragments (by Wikipedia Entities). In *CIKM '10*. ACM, 1625–1628.

[12] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *ACL '05*. 363–370.

[13] Luis Galárraga, Geremy Heitz, Kevin Murphy, and Fabian M. Suchanek. 2014. Canonicalizing Open Knowledge Bases. In *CIKM '14*. ACM, 1679–1688.

[14] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard de Melo, and Gerhard Weikum. 2011. YAGO2: Exploring and Querying World Knowledge in Time, Space, Context, and Many Languages. In *WWW '11*. ACM, 229–232.

[15] Raphael Hoffmann, Congle Zhang, and Daniel S. Weld. 2010. Learning 5000 Relational Extractors. In *ACL '10*. Association for Computational Linguistics, 286–295.

[16] Qi Li and Heng Ji. 2014. Incremental Joint Extraction of Entity Mentions and Relations. In *ACL '14*. ACL, 402–412.

[17] Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. Open Language Learning for Information Extraction. In *EMNLP-CONLL '12*. ACL.

[18] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL '09*. ACL, 1003–1011.

[19] Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *TACL* 2 (2014), 231–244.

[20] Dat Ba Nguyen, Johannes Hoffart, Martin Theobald, and Gerhard Weikum. 2014. AIDA-light: High-Throughput Named-Entity Disambiguation. In *Proceedings of the Workshop on Linked Data on the Web, LDOW '14*. CEUR-WS.org.

[21] Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. 2016. J-NERD: Joint Named Entity Recognition and Disambiguation with Rich Linguistic Features. *TACL* 4 (2016), 215–229.

[22] Joakim Nivre and Johan Hall. 2005. Maltparser: A Language-Independent System for Data-Driven Dependency Parsing. In *Proceedings of the 4th Workshop on Treebanks and Linguistic Theories*. 13–95.

[23] Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. 2013. Joint Inference of Entities, Relations, and Coreference. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction, AKBC '13*. ACM, 1–6.

[24] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *WWW '07*. ACM, 697–706.

[25] Fabian M. Suchanek, Mauro Sozio, and Gerhard Weikum. 2009. SOFIE: A Self-organizing Framework for Information Extraction. In *WWW '09*. ACM, 631–640.

[26] Charles A. Sutton and Andrew McCallum. 2012. An Introduction to Conditional Random Fields. *Foundations and Trends in Machine Learning vol. 4* (2012), 267–373.