TAC KBP 2016 Cold Start Slot Filling and Slot Filler Validation Systems by IRT SystemX

Rashedur Rahman, Brigitte Grau, Sophie Rosset, Yoann Dupont, Jérémy Guillemot, Christian Lautier, Wilson Fred

TAC KBP 2016 Cold Start Slot Filling and Slot Filler Validation Systems by IRT SystemX

Rashedur Rahman (1,2,3), Brigitte Grau (2,3,4), Sophie Rosset (2,3), Yoann Dupont (5), Jérémy Guillemot (1), Olivier Mesnard (1), Christian Lautier (5), Wilson Fred (5)

(1) IRT SystemX (2) LIMSI, CNRS (3) University Paris Saclay (4) ENSIIE (5) Expert System France
first.last@irt-systemx.fr, first.last@limsi.fr, first.last@temis.com

Abstract

This paper describes the participation of IRT SystemX at TAC KBP 2016, for the two tracks, CSSF and SFV (filtering and ensemble). We have submitted 4 runs for each track of SFV which are our first submission and this submissions are applicable for only cold start monolingual English SF/KB runs (for both filtering and ensemble). The classifier models we use for SFV track are the same for both filtering and ensemble task.

1 Introduction

This year IRT SystemX participates at TAC KBP evaluation task for two tracks: cold start monolingual English slot filling (English CSSF) and SFV (monolingual English). We submit three runs for CSSF and four runs for SFV filtering and ensemble. Our SF system first processes the collection in order to build a knowledge graph based on NER, sentence splitting, relation extraction and entity linking. We then perform SF query in this graph for collecting the candidate fillers that are submitted to a binary classifier to decide if they are correct or wrong by extracting features from the knowledge graph.

Our SFV system is also developed based on a binary classifier that uses some voting, linguistic and global knowledge features to validate a slot filler by analyzing the information provided in the SF response and the knowledge graph.

We incorporate a common technique for both tasks (SF and SFV) which is community graph based relation validation (**?**). Let, a graph $G = (V, E)$, query relation (slot) R_q, query entity $v_q \epsilon V$, candidate filler-entities $V_c = \{v_{c1}, v_{c2}, \ldots, v_{cn}\} \epsilon V$ where $R_q = e(v_q, v_c) \epsilon E$. The candidate list is generated based on the extracted relations. Suppose other semantic relations $R_o \epsilon E$ where $R_o \neq R_q$. We define the task to classify whether a filler-entity c of C_v is correct or wrong for a query relation (R_q) by analyzing the communities of query entity and candidate fillers where a community is built with a set of entities which are mostly inter-related.

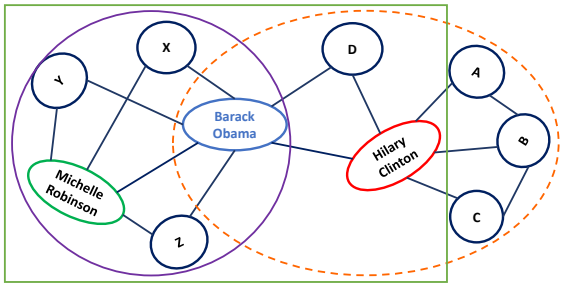

Figure 1: Community graph

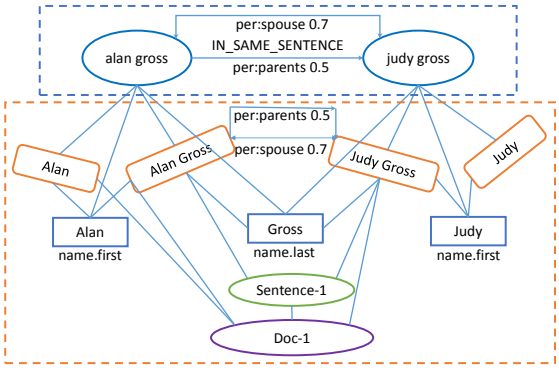

Figure 2: Knowledge graph

Fig. 1 shows an example of community based relation validation task where the query entity, type and slot name are *Barack Obama*, *person* and *spouse* accordingly. The slot filler candidates are *Michelle Robinson* and *Hilary Clinton* that are linked to *Barack Obama* by *spouse* relation hypothesis. The communities of *Barack Obama* (green rectangle), *Michelle Robinson* (purple circle) and *Hilary Clinton* (orange ellipse) are constructed by *in_same_sentence* relation which means the pair of entities are mentioned in the same sentences in the texts. We want to classify *Michelle Robinson* as the correct slot filler based on community analysis.

2 Corpus processing

This section describes some preprocessing on the KBP evaluation corpus that we use in our SF and SFV systems.

2.1 Named Entity Recognition and Classification

We used Expert System France's proprietary framework *Luxid*[1] to extract named entities in text. Before extracting NEs, Luxid first uses the XeLDA framework (**?**) to process input text up to the POS tagging, which provides a rich morphosyntactic analysis. NER is then done by rules using XeLDA as the most basic information and outputs structured NEs. For example, the system will find the person *President B. Obama* with the following components: title=*president*, firstName=*B.*, lastName=*Obama*. The biggest problem in KBP tasks being silence, improving the recall of the NER was the first step. To this effect, we extracted various lexica: a location lexicon from Geonames (**?**) and a first name lexicon using Yago3 (**?**) and removed entries that were too ambiguous or too noisy. We removed entries that were present in wiktionary's frequency lists[2] stripped of already known entities, this allows to filter the most ambiguous words. Then, we checked if they were ambiguous with common nouns. This process also allowed us to generate a list of terms that are ambiguous between different types of entities (for example *Paris* can be both a first name and a location) that would be disambiguated with specific rules.

Luxid also has an annotation propagation system at document level that allows it to retrieve more entities. It would include examples such as the person *Barack Obama* being propagated to *B. Obama*.

We also had a scope problem and could not yield some relations. For example, in Luxid, the *title* of a *person* is one of its components, not an entity of its own (meaning that *president* is not extracted unless there is a more reliable entity next to it such as *B. Obama*). Since we modeled relations between entities and not components, we could not extract the relation "per:title", despite having the information. Some entities were not extracted at all, such as facilities and religion or political affiliation. We are working on those issues, but could not integrate them for KBP.

In addition we used a fusion of Luxid and Stanford NERs specially for SFV task that increased the recall. When an entity is detected by both Luxid and Stanford we accept only the Luxid detected entity because luxid gives additional information about the entity (such as, first_name and last_name components). However, we used only Luxid for detecting named entities in the CSSF system.

2.2 Relation Detection

The great variability of ways to state a fact or a relation in natural language leads us to prefer a system based on learning than a ruled based system to extract relation. On the other hand, supervised learning systems need large corpus of text with annotations of relation. The cost of such resource exceed the budget of our project and in the perspective of submission to multilingual track, we predict that distant supervision is the most rational choice.

In this section, we describe our system for extraction of relation between mentions, which uses multiR (**?**) with distant supervision (**?**).

2.2.1 Training Data

As a first step we have built a repository of facts that conforms to KBP model. Our first attempt to build such a repository relied on FreeBase (**?**) however too many types of relation were missing. The facts are now extracted from Wikidata (**?**) : we have queried Wikidata to get the most complete set of

[1] http://www.expertsystem.com/fr/

[2] en.wiktionary.org/wiki/Wiktionary:Frequency_lists

Wikidata types (and subtypes) which maps to KBP types and relation types. We then parsed a dump of Wikidata and checked every relation between two entities to test if it is conform to a KBP model. If this is the case, we insert the entities (if they do not already exist) and the relation in the KBP repository. As a second step we build a corpus of sentences which express KBP relations. We used the source texts of TAC-KBP 2014 (news, blog, forum) corpus (**?**) to automatically produce a corpus annotated with relations. We used our NER system to detect entities mentions and sentences boundaries. As soon as two entities mentioned in the same sentence are in relation in the repository of facts, we select the sentence and add it to the corpus with an annotation which reflects the fact(s) in the repository.

In run#1, we use facts from FreeBase to produce a first model for 8 relations. In run#2, we use facts from Wikidata to produce a second model for 25 relations.

We notice that about nine tenth of sentences did not express the relation. In spite of this high level of noise, we decided no to filter or enhanced the repository of facts as done in **?**).

2.2.2 Machine Learning Model

We use MultiR with distant supervision to score relation hypotheses between entity mentions. **?**) emits the hypothesis that different sentences may express different relations for the same couple of entities, such as **founder_of**(*Jobs*,*Apple Inc.*) and **ceo_of**(*Jobs*,*Apple Inc.*). His proposition combines the following:

- the extraction of relation(s) at corpus level, that is the types of relation between a given pair of entities. For the previous example, it would be **founder_of**(*Jobs*,*Apple Inc.*) and **ceo_of**(*Jobs*,*Apple Inc.*).

- the extraction of a relation at sentence level, that is identifying the relation that is expressed between two entities in a given snippet (or *none*).

The model is both a joint probability over two random variables:

- *Y*, the variable modeling the set of relation types between two entities at corpus level and,

- *Z*, the variable modeling the set of relations instances between two entities at sentence level

and a conditional probability as is defined by the equation 1.

$$p(Y = y, Z = z|x; \theta) = \frac{1}{Z_x} \prod_r \phi^{join}(y^r, z) \prod_i \phi^{extract}(z_i, x_i) \quad (1)$$

Where Z_x is a normalization factor, ϕ^{join} is an indicator function as defined in equation 2 and $\phi^{extract}$ take the form of the equation 3.

$$\phi^{join}(y^r, z) = \begin{cases} 1 \; if \; y^r = true \; \wedge \exists \; i \; : \; z_i \; = \; r \\ 0 \; otherwise \end{cases} \quad (2)$$

$$\phi^{extract}(z_i, x_i) = exp(\Sigma_j \theta_j \phi_j(z_i, x_i) \quad (3)$$

The idea is that the training is going to be constrained by facts (in this case, *y*), but allows more flexibility for latent variables Z_i that can take multiple values for a given couple of entities depending on the sentence.

The dependencies between *x* and *y* are shown in figure 3.

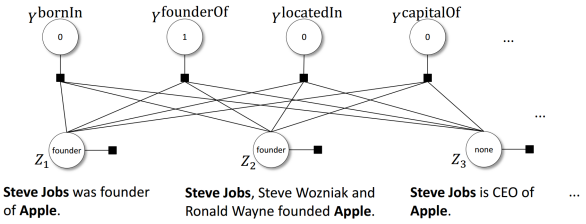

Figure 3: dependencies between variables Y and Z

2.2.3 Training

Training is done inline, with iteration on each tuple i.

- for the set of all sentences that mentions the tuple, zi' is computed as the most likely z (without taking into account the fact yi). Then a most likely value yi' is computed taking into account dependencies.

- If yi' and yi differs, we must slightly change the model: zi* is computed taking into account yi as a constraint. We alter the model of delta, as the difference between features for zi* and feature for zi'.

2.2.4 Features

We used most features mentioned by **?**) with some some differences. We used lemmatization, as **?**) did not use it. For each kind of feature, we used three variants: one using the word themselves, the second one using their POS tags and the last one their lemmas.

We did not use the combination features in our system as **?**) did. We think that is the main reason why we cannot manage to discriminate relations.

We also added different filters for words between two entities, such as nouns or verbs.

We were in the process of adding dependency based features (dependency path, words, lemmas, filtered words, etc.), but did not manage to integrate them in time for the runs.

2.2.5 Inference

We did not use MultiR to infer relations at corpus level (i.e. Y), rather only to learn how to infer values at sentence level (i.e. Z). The objective of our experiment was to consider a great set of different features at mention level and to study how to combine them to produce the best hypothesis at entity level in SFV.

2.3 Knowledge Graph and Community Graph Generation

We generate a knowledge graph as illustrated in Fig.2 after applying named entity recognition, sentence splitting and relation extraction on the evaluation corpus. The knowledge graph represents the documents, sentences, mentions and entities as nodes and relations among these are denoted by edges. The edge between two entities also holds the MultiR relation hypothesis (found at mention level) and the confidence score.

We also create a community graph (Fig. 1) based on the entity level information in the knowledge graph. Since the community graph is constructed based on the knowledge graph, the semantics are maintained in the community graph. We include *person, location and organization* typed entities as the community members in our community-graph-based analysis.

In the knowledge graph multiple mentions of the same entity (found in the same document) are linked to a common entity node. In many cases an entity is mentioned in different documents in various forms (for example, *Barack Obama, President Barack Obama, President Obama* etc) that create redundant entity nodes in the knowledge graph. We detect such kind of entity nodes based on community analysis and consolidate them into a single entity node by keeping references of the mentions, sentences and documents.

3 Global Knowledge Graph Features

We assume that a correct filler-entity of a SF query should be a strong member in the community of the query entity and such community can be extracted from the texts by extracting semantic relations and/or based on their existences in the same sentences. We hypothesize that the *network density* (eq. 4) of a community of a correct filler-entity with the query entity should be higher than a community of an incorrect filler-entity with the query entity. In Fig. 1 the community of *Michelle Robinson* with *Barack Obama* is more dense than the community of *Hilary Clinton*.

$$\rho_{network} = \frac{number\ of\ existing\ edges}{number\ of\ possible\ edges} \tag{4}$$

$$cosine\ similarity = \frac{|X \cap Y|}{\sqrt{|X||Y|}} \tag{5}$$

where, X and Y are the set of community-members of query and filler entity accordingly

$$MI(X,Y) = H(X) + H(Y) - H(X,Y) \tag{6}$$

$$\text{where, } H(X) = -\sum_{i=1}^{n} p(x_i)\ log_2(p(x_i)),$$

X and Y are the communities of query and filler entity accordingly
and p(x) refers to the probability of centrality degree of a community-member

Eigenvector centrality (**?**) measures the influence of a neighbor node to measure the centrality of a node in a community. We quantify the influence of the candidate fillers in the community of a query entity by calculating the absolute difference between the eigenvector centrality scores of the query entity and a filler entity. We hypothesize that the difference should be smaller for a correct filler than an

incorrect filler. We also hypothesize that the *mutual information* (eq. 6) and *similarity* (eq. 5) between the community of a correct filler and the community of the query entity should be higher than an incorrect filler. The community of an entity (query entity or a candidate filler-entity) is expanded up to level 3 for measuring the eigenvector centrality and mutual information.

4 CSSF System Overview

The system we used for CSSF works in two steps.

The first step extracts relation hypotheses related to the query entity mentions at sentence level based on their respective types, following the KBP model. In other words, for one given couple of entity mentions, we only generate as hypotheses the subset of the KBP relations that takes the respective mention types as argument.

The second step consolidates relation hypotheses at mention level to find a relation candidate at entity level, using graph based features to decide which is the best one between a given couple of entities. We first create entities as clusters of entity mentions, some being merged and others discarded in the process. We then use a trained classifier to rank the set of relations hypotheses and select the most likely one. A set of features was used in the classifier: MultiR score of hypothesis, frequency of hypothesis, centrality of nodes in the graph.

4.1 Runs

Three runs were submitted. Both run#1 and #2 are based on a 25-relations detection module. Run#1 makes use of a light version of the consolidation step while Run#2 make use of the complete one except for the centrality computation that was made at rank 1 only. Run#3 is based on a very simple 8-relations detection module.

Our system got few correct answers. This can be explained by many factors: the low performance of our first processing stage (entity recognition, classification of relation between mentions) and the no completions of some work (our model take into account only 25 relations among 43 from the KBP model). The main benefit of this work was to setup a global architecture for a KBP system and we plan to focus on enhancing every parts in the future.

4.2 Perspectives

We plan to improve our system by first analyzing the errors using a finer and finer grain: starting from NER up to the relation hypothesis at entity level. We also plan to include more features in our relation extraction system, starting with dependency-based features. Improving the distant learning procedure is planned as well: first, by using the pseudo-relevance relation feedback described by **?**) and studying deeper the role of negative examples. We plan to reduce the silence also, by extracting entities such as religion and political parties, as well as including coreference resolution in our system.

5 SFV System Overview

We develop our SFV system based on validating relation between the query entity and filler value by analyzing the SF system responses (relation provenance text, filler, system and confidence scores) and knowledge at the corpus level. We use voting, linguistic and global knowledge based features for SFV task. The voting features include filler, source-document, system credibility and confidence score. Basically we build classifier models for validating relations between a query and filler entity. We use the same classifier models for both SFV filtering and ensemble tasks.

Our SFV system basically contains three levels: 1. input file processing 2. feature extraction and 3. binary classification. Figure 4 illustrates the different levels of the SFV system.

Input file processing: all the system responses (input for SFV task) are merged into a single file and the responses are grouped into individual files regarding the query ids.

Feature extraction: at this level, we generate a feature vector for each response of a query by analyzing the relation provenance text, system-ids, document-ids and filler values.

Binary classification: finally each response is classified as correct or wrong by using a pre-trained classifier.

Our system decides the best filler for a single-valued slot based on the confidence score resulted by the classifier model. Moreover, the redundant fillers are resolved by simple string matching technique.

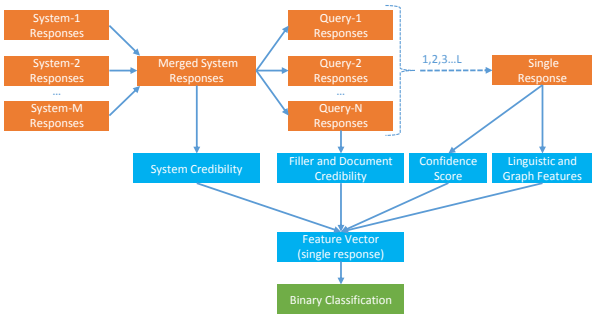

Figure 4: Slot Filler Validation System

5.1 SFV Features

In our SFV system we use three groups of features for training the relation validation classifier models and validating responded slot fillers. The feature groups are linguistic, graph and voting (including confidence score) that are briefly discussed below.

5.1.1 Linguistic Features

Usually the relation between a pair of entities is expressed by texts at the sentence level. We extract the corresponding sentence(s) of a slot filler response and perform some linguistic analysis to decide if the relation holds or not. Our linguistic analysis includes semantic and syntactic feature extraction. We analyze the seed words for characterizing the semantic of a relation. On the other hand, we use some dependency pattern based features for SFV task that includes dependency pattern length, clause detection and dependency pattern edit distance.

5.1.2 Graph Features

We analyze the community of a query entity and the candidate filler entities to decide if they support the claimed relation or not. We measure some centrality based on the community analysis as discussed in Section 3. Additionally the similarity and density of the communities are calculated as the SFV features.

5.1.3 Voting and Confidence Scores

Our SFV system takes into account some voting features by calculating the credibility of the filler values, reference documents and systems based on all the responses of a query. Moreover, we use the confidence score as a feature given by the system for a slot filling response.

5.2 Training Data and Machine Learning Models

We prepared the training data by processing the assessments of cold start slot filling responses of 2014 and 2015. Basically our system learns the trigger words and dependency patterns by analyzing the relation provenance texts of 2014 SF responses. We use the assessment data of 2015 slot filling responses for training four classifier models. These models validate a relation that is claimed by a SF response (query entity, filler value and the relation provenance text) as correct or wrong.

Basically we extract the text-snippets based on the relation provenance offset of the slot filling responses. Then the snippets are separated into two categories: positive and negative according to their relation provenance assessment. If the assessment of relation provenance offset is correct (C) (when the filler is correct or inexact) for a slot filling response the relation provenance snippet is considered as a positive snippet. Otherwise, the snippet is counted as negative. Then we generate the feature vectors of the slot filling responses and build classifier models. We notice that around $2,000$ round-1 SF queries of 2015 KBP CSSF were assessed by NIST where around 50% of the queries were responded by both correct and wrong fillers. We compile our experimental data set from the responses of these queries. We also notice that there are a lot of responses that have the same feature vector. Therefore, the duplicate vectors have been removed from the dataset. Moreover, our system is not able to extract linguistic features for some of the slot filling responses. We classify these responses by using only the features based on voting and confidence score. One of the classifier models we build by using voting, linguistic and graph features together though graph features are extracted from very few CSSF/KB responses. The classifier assigns mean values for the missing attributes in the feature vectors for which graph features are not available.

We train the Random Forest (**?**) classifier in Weka (**?**) to build 4 models regarding different feature sets. Model-1 includes voting, linguistic and community analysis. Model-2 excludes the community based features of Model-1. On the other hand, Model-3 and Model-4 use only the linguistic and voting fea-

tures accordingly.

5.3 Experiment and Results

We have submitted 4 SFV (filtering) and 4 SFV (ensemble) runs according to the classifier models described in Section 5.2. This submissions are for only Cold Start monolingual English SF/KB runs.

We evaluate our classifier models by measuring the performances of relation validation on a test corpus, which is a part of 2015 data. Table 1 shows the statistics of training and test instances for relation validation task and Table 2 depicts the precision, recall and F-score regarding the classification task. We observe that linguistic features improves the F-score significantly over the voting features. The graph features does not improve the classification performance significantly over linguistic features because our system is not able to extract graph features for all the SF responses. Graph features are extracted for a limited number of responses that counts around $5,000$ responses from 260 queries.

Model	Training	Test
voting	25,390	6,348
linguistic	26,280	6,552
voting + linguistic	26,280	6,552
voting + linguistic + graph	26,280	6,552

Table 1: Statistics of training and test data set for different models

Feature Set	P	R	F
voting	93.1	93.1	93.1
linguistic	83.3	83.8	83.4
voting + linguistic	94.4	94.4	94.4
voting + linguistic + graph	94.5	94.5	94.5

Table 2: Classification performances of different models (in %)

We have submitted 4 runs for SFV (ensemble) task that use different feature sets for validating relations: Run#1(voting + linguistic + graph), Run#2(voting + linguistic), Run#3 (linguistic) and Run#4(voting). Figure 5 depicts the official score of SFV (ensemble) task. Run#2 achieves the highest F-score (24.79) among 4 runs. We compare this score to the CS KB/SF runs by different systems as shown

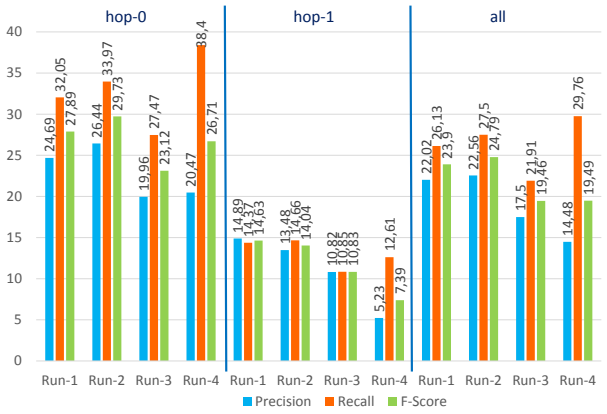

Figure 5: SFV (ensemble) official score

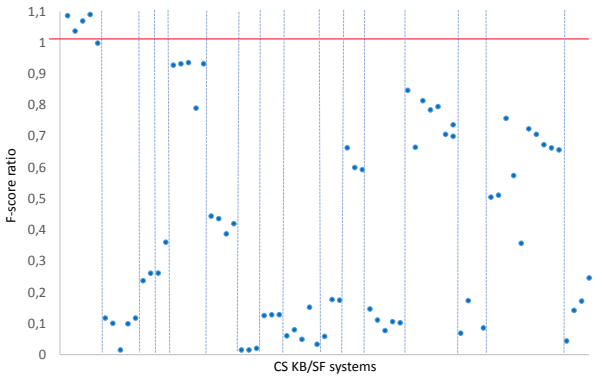

Figure 6: Ratio of CS KB/SF F-scores to IRT SystemX SFV (ensemble) submission (Run#2)

in Figure 6. We calculate the ratio of F-score of CS KB/SF runs to our best SFV ensemble run. Only four CS KB/SF runs obtain higher F-score than our SFV ensemble run.

6 Conclusion

In this paper we present the TAC KBP2016 CSSF and SFV system by IRT SystemX. We apply a graph based relation validation method for selecting the correct slot filler(s) among several candidates. The SF system uses distant supervision for extracting relations by using MultiR. Our current SF system is limited to extracting 25 KBP relations and has to be improved to extracting all the KBP relations defined by TAC. On the other hand, the SFV system builds some binary classification models based on several features that include global knowledge, linguistic and voting features. We submit SFV runs for

the first time ever. Our current SFV system is not efficient enough to filter out redundant fillers which has to be improved in the future.

References

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM.

Phillip Bonacich and Paulette Lloyd. 2001. Eigenvector-like measures of centrality for asymmetric relations. *Social networks*, 23(3):191–201.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2013. The weka data mining software: An update; sigkdd explorations, volume 11, issue 1, 2009. *Reproduced with permission of the copyright owner. Further reproduction prohibited without permission*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 541–550, Stroudsburg, PA, USA. Association for Computational Linguistics.

Andy Liaw and Matthew Wiener. 2002. Classification and regression by randomforest. *R News*, 2(3):18–22.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. 2014. Yago3: A knowledge base from multilingual wikipedias. In *7th Biennial Conference on Innovative Data Systems Research*. CIDR Conference.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Hervé Poirier. 1999. The xelda framework. In *Baslow workshop on Distributing and Accessing Linguistic Resources*, pages 33–38. Sheffield.

Rashedur Rahman, Brigitte Grau, and Sophie Rosset. 2016. Graph based relation validation method. In *International Conference on Knowledge Engineering and Knowledge Management, EKAW*.

Mihai Surdeanu and Heng Ji. 2014. Overview of the english slot filling track at the tac2014 knowledge base population evaluation. In *Proc. Text Analysis Conference (TAC2014)*.

Bernard Vatant and Marc Wick. 2012. Geonames ontology.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Wei Xu, Raphael Hoffmann, Le Zhao, and Ralph Grishman. 2013. Filling knowledge base gaps for distant supervision of relation extraction. In *ACL (2)*, pages 665–670.