A Simple Distant Supervision Approach for the TAC-KBP Slot Filling Task

Mihai Surdeanu, David McClosky, Julie Tibshirani, John Bauer,
Angel X. Chang, Valentin I. Spitkovsky, Christopher D. Manning
Computer Science Department
Stanford University, Stanford, CA 94305
{mihais,mcclosky,jtibs,horatio,angelx,vals,manning}@stanford.edu

Abstract

This paper describes the design and implementation of the slot filling system prepared by Stanford's natural language processing group for the 2010 Knowledge Base Population (KBP) track at the Text Analysis Conference (TAC). Our system relies on a simple distant supervision approach using mainly resources furnished by the track organizers: we used slot examples from the provided knowledge base, which we mapped to documents from several corpora, i.e., those distributed by the organizers, Wikipedia, and web snippets. Our implementation attained the median rank among all participating systems.

1 Introduction

This paper describes the slot filling system prepared by Stanford's natural language processing (NLP) group for the Knowledge Base Population (KBP) track of the 2010 Text Analysis Conference (TAC). Our system adapts the distant supervision approach of Mintz et al. (2009) to the KBP slot filling context. We: (a) extract slot (or relation) instances from a knowledge base; (b) map these instances to sentences in document collections; and (c) train a statistical model using these examples. However, there are several significant differences between our approach and that of Mintz et al. (2009): (a) we use mainly the resources provided by the task organizers, i.e., Wikipedia infoboxes and the KBP corpus, instead of Freebase;[1] (b) we implement a one-to-many mapping from infobox elements to KBP slots since Wikipedia infoboxes do not align with the KBP slot types; and (c) we couple the slot extraction component with an information retrieval (IR) system to accommodate the large document collection provided.

Figure 1 summarizes our system's architecture. For clarity, we present two distinct execution flows: one for training the slot classifier, and one for evaluating the entire system.

2 Training

2.1 Mapping Infobox Fields to KBP Slot Types

We used the Wikipedia infoboxes provided by the task organizers as our source of distant supervision. However, these infoboxes contain arbitrary fields that map to none, one, or more KBP slot types. For example, the infobox field `University:established` maps one-to-one to the KBP slot type `org:founded`. But the infobox field `Writer:children` maps to either zero, one, or more `per:children` slots. For example, we disregard the infobox field (`Writer:children`, "John Steinbeck", "3") because the text "3" cannot contain a name. On the other hand, we map the field (`Writer:children`, "Mark Twain", "Langdon, Susie") to two KBP slots: (`per:children`, "Mark Twain", "Langdon") and (`per:children`, "Mark Twain", "Susie"). In the same vein, we map the infobox field (`University:address`, "MIT", "Cambridge, Mass.") to two KBP fields: (`org:city_of_headquarters`, "MIT", "Cambridge") and

[1] `http://www.freebase.com/`

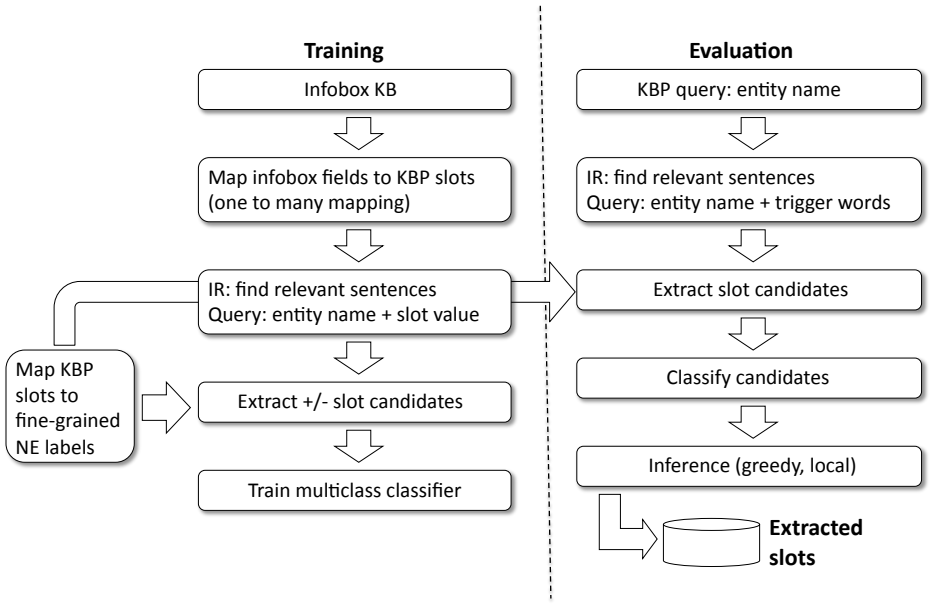

Figure 1: Architecture of the slot filling system.

(`org:stateorprovince_of_headquarters`, "MIT", "Mass."), and the field (`Politician:office`, "Barack Obama", "President of the United States") to two slots: (`per:title`, "Barack Obama", "president") and (`per:employee_of`, "Barack Obama", "United States"). All these conversions are implemented with deterministic rules, customized for each infobox field type.

2.2 Retrieving Relevant Sentences

We retrieve sentences that contain the previously generated slot instances by querying all our document collections with a set of queries, where each query contains an entity name and one slot value, e.g., "Barack Obama" `AND` "United States" for the previous example. From the documents retrieved, we keep only sentences that contain both the entity name and the slot value. We retrieved up to 100 sentences per entity.

We used three document collections for sentence retrieval during training:

1. The official document corpus provided by the task organizers.

2. Snippets of Wikipedia articles from the 2009 TAC-KBP Evaluation Reference Knowledge Base. These snippets are often prefixes of the complete articles. Nevertheless, they are extremely useful because they correspond to the Wikipedia pages of the infoboxes we used for distant supervision, so we expect most of the slots to have a match in these texts. To maximize the number of relevant sentences extracted, in this corpus we employed a shallow and fast coreference resolution with replacement: for person pages, we replaced all animate pronouns, possessive and otherwise, with the article's title. For organizations, we did the same for inanimate pronouns and also searched for possible abbreviations of the article title. For example, for the article titled "International Business Machines", we replaced all instances of "IBM" and "I.B.M." with the full title. For both people and organizations, we replaced non-empty subsequences of the article title with the complete title, for improved uniformity.

3. A complete Wikipedia snapshot from November 2008.

Note that, with the exception of the second corpus,

we did not use coreference resolution in this shared task, requiring an exact string match between entity names and slot values in text, to consider a sentence as relevant for a given slot.

2.3 Extracting Positive and Negative Slot Candidates

Following Mintz et al. (2009), we pretend that all sentences containing an entity name and a known slot value are positive example for the corresponding slot types.[2] We consider as negative slot examples all entity mentions extracted by a named entity recognizer that do not match a known slot value. Additionally, these mentions must appear in the same sentence with the entity whose slots are currently modeled, and have a type known to match a KBP slot. For example, if the current entity modeled is "Apple" in the sentence "As Apple launched the first subscription app for iPad with News Corp., Google announced catchup steps for Android.", the ORGANIZATION mentions "News Corp." and "Google" become negative slot examples because they do not match a known slot for "Apple". The mapping from KBP slot types to named entity (NE) labels was performed manually, and is released with this paper.[3]

We extract slot candidates using the Named Entity Recognizer (NER) from the Stanford CoreNLP package.[4] We extended the NER with a series of labels specific to KBP, e.g., countries, provinces, diseases, religions, etc. All these additional classes were recognized using static lists manually built by mining the Web. These lists are available for download at: `http://www.surdeanu.name/mihai/kbp2010/ner_extensions.txt`.

This process generated approximately 190K positive slot examples and 900K negative examples.

2.4 Training the Slot Classifier

We trained the slot classifier using a single multi-class logistic regression with L_2 regularization. To control for the excessive number of negative examples, we subsampled them with a probability of 0.5, i.e., we used only half of the negative examples.

The classifier features were inspired by previous work (Surdeanu and Ciaramita, 2007; Mintz et al., 2009; Riedel et al., 2010) and include:

- Information about the entity and slot candidate, e.g., the NE label of the slot candidate, and words included in the slot candidate.
- Surface features: textual order of the entity and slot candidate, number of words between entity and slot, words immediately to the left and right of the entity and slot, the NE mentions seen between the entities and slots, and, finally, words that appear between the entity and slot candidate.
- Syntactic features: the path from an entity to the slot in the constituent parse tree, and dependency path between entity and slot (both lexicalized and unlexicalized). The constituent trees and dependency graphs were built using the parser from the Stanford CoreNLP package.

3 Evaluation

3.1 Retrieving Relevant Sentences

At run-time we retrieve candidate sentences using, for each entity, a set of queries that couple the entity name with specific trigger words for each slot type. For example, for the `org:alternate_names` slot type, we use trigger words such as "called", "formerly" and "known as". These lists of trigger words were built manually and are available in their entirety.[5]

In addition to the three document collections mentioned in the previous sub-section, during evaluation we used also a web-based corpus. This corpus was constructed as follows. For each evaluation entity we constructed a set of web queries by concatenating the entity name with each trigger word (or phrase) from the above list. For each query, we retrieved up to 100 snippets from Google.[6] We created one separate document for the results of each query.

[2] This assumption is often wrong. For example, if we see that a conference was held in Austin, TX, we will learn that host cities tend to be capitals, which neither follows logically, nor happens to be true, in general.

[3] `http://www.surdeanu.name/mihai/kbp2010/slots_to_ne.txt`

[4] `http://nlp.stanford.edu/software/corenlp.shtml`

[5] `http://www.surdeanu.name/mihai/kbp2010/trigger_words.txt`

[6] `http://www.google.com/`

As in the training setup, we retrieved up to 100 sentences per entity from the other three static document collections.

3.2 Candidate Extraction, Classification, and Inference

During evaluation, we consider as slot candidates all NEs that have a type known to correspond to a slot type and that appear in the same sentence with the evaluation entity. For each slot candidate we pick the label proposed by our multiclass slot classifier independently of the other candidates. In case of conflicts, i.e., multiple extractions proposed for the same slot, we select the candidate with the highest classifier confidence. In this work, we treat `single` and `list` slots identically, i.e., we output at most one extraction per slot.

Note that there is a significant difference between our approach and previous distantly-supervised work on relation extraction (Mintz et al., 2009; Riedel et al., 2010). Both these works model slots (or relations), where each slot aggregates *all* mentions of the same value, whereas we model each slot mention individually. To produce a KBP-compliant output, we merge different mentions with the same value as follows: (a) we sum all probability scores proposed by the slot classifier for all mentions with the same value; (b) we pick the label with the highest score; and (c) if the overall score of this label is larger than 0.75 we report the classifier label, otherwise we discard the slot.

4 Results

We report scores from out development setup in Table 1. For this experiment we used two thirds of the infoboxes as training data, leaving one third for testing. We retrieved candidates from the three document collections used for training. Note that the scores in the table are likely to be more conservative than those in previous works (Mintz et al., 2009; Riedel et al., 2010), because we report results for each slot mention, rather than for entire slots or relations. In other words, in our evaluation each individual mention is scored separately, whereas both Mintz et al. (2009) and Riedel et al. (2010) consider a slot as correct if its mentions are classified correctly on average.

Overall, our system obtains a F1 score of 56.7 in the development set. This value is slightly higher than the scores obtained by Mintz et al. (2009) and Riedel et al. (2010) in comparable experiments. As the table indicates, some of the slot types can be extracted with high accuracy, e.g., `per:date_of_birth`, whereas others are considerably more difficult, e.g., `org:top members/employees`.

Nevertheless, our KBP scores on the official test partition are low. Our system obtained a F1 score of 14.12 (10.54 precision and 21.41 recall) when the web snippet collection is used, and 12.25 F1 (24.07; 8.22) without the web snippets. The former configuration attained the median rank among all participating systems.

5 Conclusions and Future Work

This paper introduced a simple application of the distant supervision approach to the TAC-KBP slot filling task. With the exception of the pre-existing slot classifier, this entire system was created in approximately two calendar weeks. Due to this tight development schedule, several important issues were left unexplored. We plan to address the most important ones in future work:

- We suspect that the drop between our development scores and the official KBP scores is caused by the low recall of the IR module. We will focus on improving our IR component, i.e., develop a distributed implementation capable of processing more sentences per entity. We will also improve our trigger word detection strategy (see, e.g., Chapter 23 in (Jurafsky and Martin, 2009)).

- We will investigate the contribution of syntactic and discourse processing tools to this task, e.g., what is the improvement if true coreference resolution for entity names and slot values is used? Which syntactic dependency representation is best for slot extraction?

- Riedel et al. (2010) showed that the assumption that all sentences that contain an entity name and known slot values are positive examples for the relevant slot type is wrong, especially

Label	Correct	Predicted	Actual	P	R	F1
NIL	268,085	289,135	295,590	92.7	90.7	91.7
org:city_of_headquarters	5,835	9,040	7,514	64.5	77.7	70.5
org:country_of_headquarters	2,851	4,638	3,725	61.5	76.5	68.2
org:founded	3,896	8,199	6,662	47.5	58.5	52.4
org:parents	1,158	2,292	2,525	50.5	45.9	48.1
org:top_members/employees	1,282	3,067	3,596	41.8	35.7	38.5
per:city_of_birth	1,799	3,920	3,252	45.9	55.3	50.2
per:country_of_birth	1,984	4,122	3,204	48.1	61.9	54.2
per:date_of_birth	3,938	5,427	4,362	72.6	90.3	80.5
per:member_of	1,771	3,018	2,887	58.7	61.3	60.0
per:title	1,714	3,364	3,054	51.0	56.1	53.4
...						
Total	37169	68822	62367	54.0	59.6	56.7

Table 1: Results of the distantly supervised system that two thirds of the KB as training data and the remaining third of the KB as testing. We score only mentions of slots that appeared at least once in the underlying document collections. The first rows of the table shows results for the NIL label (i.e., entity mentions that are not known to be slots), followed by the five most common slots for organization and person entities. The final row is an aggregate of the overall scores for all slots.

in non-Wikipedia collections. We will investigate models capable of discriminating between true and false positive slot examples.

References

D. Jurafsky and J.H. Martin. 2009. Speech and Language Processing (2nd Edition). Prentice-Hall, Inc.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. 2009. Distant supervision for relation extraction without labeled data In *ACL-IJCNLP*.

S. Riedel, L. Yao, and A. McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML/PKDD*.

M. Surdeanu and M. Ciaramita. 2007. Robust Information Extraction with Perceptrons. In *ACE*.