Confidence Estimation for Knowledge Base Population

Xiang Li
Computer Science Department
New York University
xiangli@cs.nyu.edu

Ralph Grishman
Computer Science Department
New York University
grishman@cs.nyu.edu

Abstract

Information extraction systems automatically extract structured information from machine-readable documents, such as newswire, web, and multimedia. Despite significant improvement, the performance is far from perfect. Hence, it is useful to accurately estimate confidence in the correctness of the extracted information. Using the Knowledge Base Population Slot Filling task as a case study, we propose a confidence estimation model based on the Maximum Entropy framework, obtaining an average precision of 83.5%, Pearson coefficient of 54.2%, and 2.3% absolute improvement in F-measure score through a weighted voting strategy.

1 Introduction

Despite significant progress in recent years, Information Extraction (IE) technologies are still far from completely reliable. Errors result from the fact that language itself is ambiguous as well as methodological and technical limitations (Gandrabur et al., 2006). Therefore, evaluating the probability that the extracted information is correct can contribute to improve IE system performance. Confidence Estimation (CE) is a generic machine learning rescoring approach for measuring the probability of correctness of the outputs, and usually adds a layer on top of the baseline system to analyze the outputs using additional information or models (Gandrabur et al., 2006). There is previous work in IE using probabilistic and heuristic methods to estimate confidence for extracting fields using a sequential model, but to the best of our knowledge, this work is the first probabilistic CE model for the multi-stage systems employed for the Knowledge Base Population (KBP) Slot Filling task (Section 2).

The goal of Slot Filling (SF) is to collect information from a corpus of news and web documents to determine a set of predefined attributes ("slots") for given person and organization entities (Ji et al., 2011a) (Section 3). Many existing methodologies have been used to address the SF task, such as Distant Supervision (Min et al., 2012) and Question Answering (Chen et al., 2010), and each method has its own strengths and weaknesses. Many current KBP SF systems actually consist of several independent SF pipelines. The system combines intermediate responses generated from different pipelines into final slot fills. Since these intermediate outputs may be highly redundant, if confidence values can be associated, it will definitely help re-ranking and aggregation. For this purpose, we require comparable confidence values from disparate machine learning models or different slot filling strategies.

Robust probabilistic machine learning models are capable of accurate confidence estimation because of their intelligent handling of uncertainty information. In this paper, we use the Maximum Entropy (MaxEnt) framework (Berger et al., 1996) to automatically predict the correctness of KBP SF intermediate responses (Section 4). Results achieve an average precision of 83.5%, Pearson's r of 54.2%, and 2.3% absolute improvement in final F-measure score through a weighted voting system (Section 5).

2 Related Work

Confidence estimation is a generic machine learning approach for measuring confidence of a given output, and many different CE methods have been used extensively in various Natural Language Processing (NLP) fields (Gandrabur et al., 2006). Gandrabur and Foster (2003) and Nguyen et al. (2011) investigated the use of machine learning approaches for confidence estimation in machine translation. Agichtein (2006) showed

Proceedings of Recent Advances in Natural Language Processing, pages 396–401,
Hissar, Bulgaria, 7-13 September 2013.

Expectation-Maximization algorithms to estimate the confidence for partially supervised relation extraction. White et al. (2007) described how a maximum entropy model can be used to generate confidence scores for a speech recognition engine. Louis and Nenkova (2009) presented a study of predicting the confidence of automatic summarization outputs. Many approaches for confidence estimation have also been explored and implemented in other NLP research areas.

There are also many previous confidence estimation studies in IE, and most of these have been in the Active Learning literature. Thompson et al. (1999) proposed a rule-based extraction method to compute confidence. Scheffer et al. (2001) utilized hidden Markov models to measure the confidence in an IE system, but they only estimated the confidence of singleton tokens. Culotta and McCallum (2004)'s work is the most relevant to our work, since they also utilized a machine learning model to estimate the confidence values for IE outputs. They estimated the confidence of both extracted fields and entire multi-field records mainly through a linear-chain Conditional Random Field (CRF) model, but their case studies are not as complicated and challenging as slot filling, since SF systems need to handle difficult cross-document coreference resolution, sophisticated inference, and also other challenges (Min and Grishman, 2012). Furthermore, to the best of our knowledge, there is no previous work in confidence estimation for the KBP slot filling task.

3 KBP Slot Filling

3.1 Task Definition

The Knowledge Base Population (KBP) track, organized by U.S. National Institute of Standards and Technology (NIST)'s Text Analysis Conference (TAC), aims to promote research in discovering information about entities and augmenting a Knowledge Base (KB) with this information (Ji et al., 2010). KBP mainly consists of two tasks: Entity Linking, linking names in a provided document to entities in the KB or NIL; and Slot Filling (SF), extracting information about an entity in the KB to automatically populate a new or existing KB. As a new but influential IE evaluation, Slot Filling is a challenging and practical task (Min and Grishman, 2012).

The Slot Filling task at *KBP2012* provides a large collection of 3.7 million newswire articles and web texts as the source corpus, and an initial KB derived from the Wikipedia infoboxes. In such a large corpus, some information can be highly redundant. Given a list of person (PER) and organization (ORG) entity names ("queries"), SF systems retrieve the documents about these entities in the corpus and then fill the required slots with correct, non-redundant values. Each query consists of the name of the entity, its type (PER or ORG), a document (from the corpus) in which the name appears, its node ID if the entity appears in the provided KB, and the slots which need not be filled. Along with each slot fill, the system should also provide the ID of the document that justifies this fill. If the system does not extract any information for a given slot, the system just outputs "NIL" without any document ID. The task defines a total of 42 slots, 26 for person entities and 16 for organization entities. Some slots are single-valued, like "per:date_of_birth", which can only accept at most a single value, while the other slots, for example "org:subsidiaries", are list-valued, which can take a list of values. Since the overall goal is to augment an existing KB, the redundancy in list-valued slots must be detected and avoided, requiring a system to identify different but equivalent strings. Such as, both "United States" and "U.S." refer to the same country. More information can be found in the task definition (Ji et al., 2010).

3.2 Baseline System Description

We use a slot filling system that has achieved highly competitive results (ranked top 2) at the *KBP2012* evaluation as our baseline. Like most SF systems, our system has three basic components: Document Retrieval, Answer Extraction, and Response Combination. Our SF system starts by retrieving relevant documents based on a match to the query name or the results of query expansion. Then our system applies a two-stage process to generate final slot fills: Answer Extraction, which produces intermediate responses from different pipelines, and Response Combination, which merges all intermediate responses into final slot fills. Answer extraction begins with document pre-processing, such as part-of-speech tagging, name tagging, and coreference resolution. Then it uses a set of 6 SF pipelines operating in parallel on the retrieved documents to extract answers. Our pipelines consist of two that use hand-coded

	PER#	ORG#	Total#	Response#
KBP2010	50	50	100	7917
KBP2011	50	50	100	14976
KBP2012	40	40	80	8989
total	140	140	280	31878

Table 1: Number of Queries and Number of Intermediate Responses from Each Year Data

patterns, two pattern-based slot fillers in which the patterns are generated semi-automatically from a bootstrapping procedure, one based on name coreference, and one distant-supervision based pipeline. The result of this stage is a set of intermediate slot responses, potentially highly redundant. Next, Response Combination validates answers and eliminates redundant answers to aggregate all intermediate responses into final slot fills, where the best answer is selected for each single-valued slot and non-redundant fills are generated for list-valued slots. More details about our KBP Slot Filling system can be found in the system description paper (Min et al., 2012).

4 Confidence Estimation Model

Our confidence estimation model is based on the Maximum Entropy (MaxEnt) framework, a probabilistic model able to incorporate all features into a uniform model by assigning weights automatically. We implement a mix of binary and real-valued features from different aspects to estimate confidence of each intermediate slot filling response under a consistent and uniform standard, incorporating four categories of features: **Response Features** extract features from the slot and the *Response* context; **Pipeline Features** indicate how well each pipeline performed previously; **Local Features** explore how *Query* and *Response* are correlated in the supporting context *Sentence*; **Global Features** detect how closely *Query* correlates with *Response* in the global context. Each specific feature in the above categories is listed in Table 2, where *Q* refers to a person or organization *Query*; *R* indicates the pipeline-generated *Response* for a particular slot of a query; and *S* represents the *Sentence* that supports the correctness of the *Response*.

5 Experiments

We have collected and merged the previous three years' KBP SF evaluation data, which consists of a total of 280 queries, and Table 1 lists the number of person and organization queries as well as the number of intermediate responses from each year. There are in total 31878 intermediate responses generated by 6 different pipelines from our SF system. We trained our CE model and measured the confidence values through a 10-fold cross-validation, so that each fold randomly contains 14 person queries and 14 organization queries with their associated intermediate responses. Then for each iteration, the CE model is trained on 9 folds and approximates the confidence values in the remaining fold, and it assigns the probability of each intermediate response being correct as confidence.

5.1 Voting Systems

To evaluate the reliability of confidence values generated by this model, we used the weighted voting method to investigate the relationship between the confidence values and the performance.

5.1.1 Baseline Voting System

Our baseline SF system applies a basic plurality voting to combine all intermediate responses to generate the final response submission. This voting system simply counts the frequencies of each response entity, which is a unique response tuple in the form <Query_ID, Slot_Name, Response_Fill>. For a single-valued slot of a query, the response with the highest count is returned as the final response fill. For the list-valued slots, all non-redundant responses are returned as the final response fills. In this basic voting system, each intermediate response contributes equally.

5.1.2 Weighted Voting System

Weighted voting is based on the idea that not all the voters contribute equally. Instead, voters have different weights concerning the outcome of an election. In our experiment, voters are all of intermediate responses generated by all pipelines, and the voters' weights are their confidence values. We set a threshold τ in this weighted voting system, where those intermediate responses with

Category	Feature	Description
Response Features	slot_name	The slot name
	slot_response_length	The conjunction of the length of *R* and the slot name
	name_response_slot	The slot requires a name as the response
Pipeline Features	pipeline_name	The name of pipeline which generates *R*
	pipeline_precision	The Precision of the pipeline which generates *R*
	pipeline_recall	The Recall of the pipeline which generates *R*
	pipeline_fmeasure	The F-measure of the pipeline which generates *R*
Local Features	sent_contain_QR	*S* contains both original *Q* and *R*
	sent_contain_ExQR	*S* contains both co-referred *Q* or expanded *Q* and *R*
	dpath_length	The length of shortest dependency path between *Q* and *R* in *S*
	shortest_dpath	The shortest dependency path between *Q* and *R* in *S*
	NE_boolean	*R* is a person or organization name in *S*
	NE_margin	The difference between the log probabilities of this name *R* and the second most likely name
	n-gram	Tri-gram context window associated with part-of-speech tags containing *Q* or *R*
	genre	The supporting document is a newswire or web document
Global Features	query_doc_num	The number of documents retrieved by *Q*
	response_doc_num	The number of documents retrieved by *R*
	co-occur_doc_num	The number of documents retrieved by the co-occurrences of *Q* and *R*
	cond_prob_givenQ	The conditional probability of *R* given *Q*
	cond_prob_givenR	The conditional probability of *Q* given *R*
	mutual_info	The Point-wise Mutual Information (PMI) of *Q* and *R*

Table 2: Features of Confidence Estimation Model

confidences that are lower than τ would be eliminated. For each response entity, this weighted voting system simply sums all the weights of the intermediate responses that support this response entity as its weight. Then for a single-valued slot of a query, it returns the response with the highest weight as the final slot fill, while it returns all non-redundant responses as the final slot fills for the list-valued slots. The maximum confidence ψ of supporting intermediate responses is used as the final confidence for that slot fill. We also set a threshold η (optimized on a validation data set), where the final slot fills with confidence ψ lower than η would not be submitted finally.

5.1.3 Results

Table 3 compares the results of this weighted voting system (with $\tau = 0$, $\eta = 0.17$) and the baseline voting system, where the responses were judged based only on the answer string, ignoring the document ID. As we can see, the weighted voting system achieves 2.3% absolute improvement in F-measure over the baseline, at a 99.8% confidence level according to the Wilcoxon Matched-Pairs Signed-Ranks Significance Test. Precision obtains 9.0% absolute improvement with only a small loss of 0.5% in Recall.

	Precision	**Recall**	**F-measure**
Baseline	0.351	**0.246**	0.289
Weighted	**0.441**	0.241	**0.312**

Table 3: Results Comparison between Baseline Voting System and Weighted Voting System

Figure 1 summarizes the results of this weighted voting system with different threshold τ settings. When τ is raised, Precision continuously increases to around 1, while Recall gradually decreases to 0.

In addition to improving overall performance, the confidence estimates can be used to convey to the user of slot filling output our confidence in individual slot fills. After the intermediate responses are combined by the above weighted voting system (setting τ and η as 0), we divide the range of confidence values (0 to 1) into 10 equal intervals (0 to 0.1, 0.1 to 0.2, and so on) and categorize these

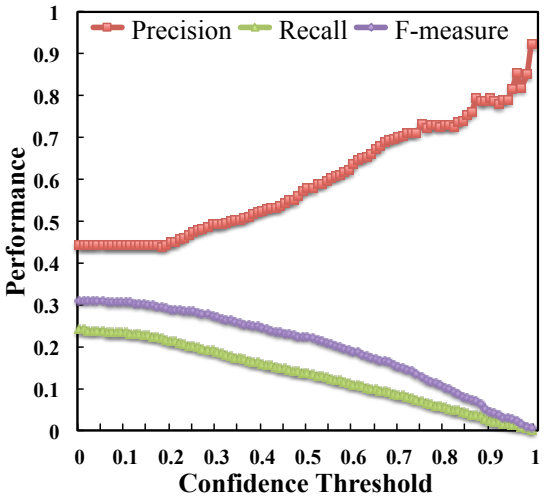

Figure 1: Impact of Threshold Settings

final slot fills by their confidence values. Then for each category, the final slot fills are scored in Precision. Figure 2 strongly demonstrates that the slot fills with higher confidence consistently generate more precise answers, indirectly validating the reliability of the confidence estimates.

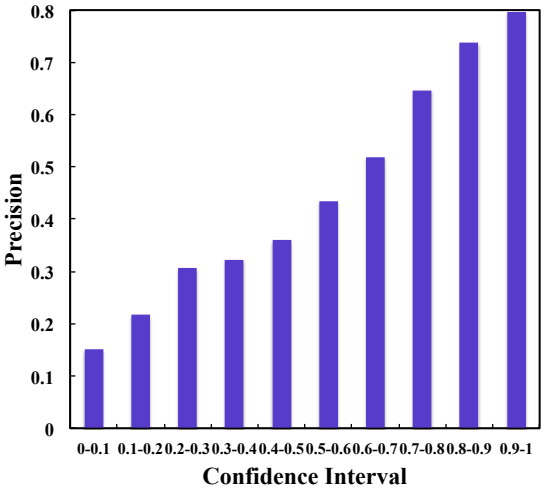

Figure 2: Performance of Confidence Intervals

5.2 Evaluation

We use another two different methods to evaluate the quality of confidence estimation in a more direct way. The first method is *Pearson's r*, a correlation coefficient ranging from -1 to 1 that measures the correlation between a confidence value and whether or not the instance is correct. It is widely used in the sciences as a measure of linear dependence between two variables. The second method is *average precision*, used in the Information Retrieval community to evaluate a ranked list. It calculates the precision at each point in the ranked list where a relevant document is found and then averages these values. Instead of ranking documents by their relevance scores, the intermediate responses are ranked by their confidence values.

	Avg. Prec	**Pearson's r**
RANKED	**0.835**	**0.542**
RANDOM	0.525	0.001
WORSTCASE	0.330	-

Table 4: Evaluation of Confidence Estimates

Table 4 shows the Pearson's r and average precision results for all intermediate responses, where RANKED ranks the responses based on their confidence values; RANDOM assigns confidence values uniformly at random between 0 and 1; WORSTCASE ranks all incorrect responses above all correct ones.

Applying the features separately, we find that *slot_response_length* and *response_doc_num* are the best predictors of correctness. *dpath_length* (the length of the shortest dependency path between query and response) is also a significant contributor. Among the features, only *NE_margin* seeks to directly estimate the confidence of a pipeline component, and it makes only a minimal contribution to the result. Overall this shows that confidence can be predicted quite well from features of the query and response, their appearance in the corpus, and prior IE system performance, without modeling the confidence of individual pipeline components.

6 Conclusion

We have presented our Maximum Entropy based confidence estimation model for information extraction systems. The effectiveness of this model has been demonstrated in the challenging Knowledge Base Population Slot Filling task, where a weighted voting system achieves 2.3% absolute improvement in F-measure score based on the confidence estimates. A strong correlation between the confidence estimates in KBP slot fills and the correctness has also been proved by obtaining an average precision of 83.5% and Pearson's r of 54.2%. In the future, further experiments are planned to investigate more elaborate models, explore more interesting feature sets, and study the contribution of each feature through a more detailed and thorough analysis.

References

Eugene Agichtein. 2006. *Confidence Estimation Methods for Partially Supervised Relation Extraction*. In Proceedings of SDM 2006.

Adam Berger, Vincent Della Pietra, and Stephen Della Pietra. 1996. *A Maximum Entropy Approach to Natural Language Processing*. Computational Linguistics, Volume 22 Issue 1, March 1996, Pages 39-71, MIT Press Cambridge, MA, USA.

Zheng Chen, Suzanne Tamang, Adam Lee, Xiang Li, Wen-Pin Lin, Javier Artiles, Matthew Snover, Marissa Passantino, and Heng Ji. 2010. *CUNY-BLENDER TAC-KBP2010 Entity Linking and Slot Filling System Description*. In Proceedings of Text Analytics Conference (TAC) 2010.

Aron Culotta and Andrew McCallum. 2004. *Confidence Estimation for Information Extraction*. In Proceedings of HLT-NAACL 2004.

Simona Gandrabur, George Foster, and Guy Lapalme. 2006. *Condence Estimation for NLP Applications*. In ACM Transactions on Speech and Language Processing, Vol. 3, No. 3, October 2006, Pages 129.

Simona Gandrabur and George Foster. 2003. *Confidence Estimation for Translation Prediction*. In Proceedings of CoNLL 2003.

Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. 2010. *Overview of the TAC2010 Knowledge Base Population Track*. In Proceedings of Text Analytics Conference (TAC) 2010.

Heng Ji, Ralph Grishman, and Hoa Trang Dang. 2011. *Overview of the TAC2011 Knowledge Base Population Track*. In Proceedings of Text Analytics Conference (TAC) 2011.

Heng Ji and Ralph Grishman. 2011. *Knowledge Base Population: Successful Approaches and Challenges*. In Proceedings of ACL 2011.

Annie Louis and Ani Nenkova. 2009. *Performance Confidence Estimation for Automatic Summarization*. In Proceedings of ACL 2009.

Bonan Min, Xiang Li, Ralph Grishman, and Ang Sun. 2012. *New York University 2012 System for KBP Slot Filling*. In Proceedings of Text Analytics Conference (TAC) 2012.

Bonan Min and Ralph Grishman. 2012. *Challenges in the TAC-KBP Slot Filling Task*. In Proceedings of LREC 2012.

Bach Nguyen, Fei Huang, and Yaser Al-Onaizan. 2011. *Goodness: A Method for Measuring Machine Translation Confidence*. In Proceedings of ACL 2011.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. *Active Hidden Markov Models for Information Extraction*. In Proceedings of IDA 2001.

Ang Sun, Ralph Grishman, Wei Xu, and Bonan Min. 2011. *New York University 2011 System for KBP Slot Filling*. In Proceedings of Text Analytics Conference (TAC) 2011.

Cynthia Thompson, Mary Califf, and Raymond Mooney. 1999. *Active Learning for Natural Language Parsing and Information Extraction*. In Proceedings of 16th International Conference on Machine Learning.

Christopher White, Alex Acero, and Julian Odell. 2007. *Maximum Entropy Confidence Estimation For Speech Recognition*. In Proceedings of ICASSP 2007.