Adding Smarter Systems Instead of Human Annotators: Re-ranking for System Combination

Suzanne Tamang
Computer Science Department
Graduate Center
City University of New York
stamang@gc.cuny.edu

Heng Ji
Computer Science Department
Graduate Center and Queens College
City University of New York
hengjicuny@gmail.com

ABSTRACT

Using a Knowledge Base Population (KBP) slot filling task as a case study, we describe a re-ranking framework in the context of two experimental settings: (1) *high transparency*; a few pipelines share similar resources that can be used to provide the developer detailed intermediate answer results; (2) *low transparency*; many systems use diverse resources, and serve as black boxes, absent of any intermediate system results. In both settings, our results show that statistical re-ranking can effectively combine automated systems, achieving better performance than the best state-of-the-art individual system (6.6% absolute improvement in F-score) and alternative combination methods. Furthermore, to create labeled data for system development and assessment, information extraction tasks often require expensive human annotators to struggle with the vast amounts of information contained within a large scale corpus. In this paper, we demonstrate the impact of our learning-to-rank framework to combine output from multiple slot filling systems to populate entity-attribute facts in a knowledge base. We show that our approach can be used to create answer keys more efficiently and at a lower cost (63.5% reduction) than laborious human annotation.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms: Experimentation, Performance

Keywords: text analysis, knowledge base population, information extraction, supervised re-ranking

1. INTRODUCTION

The goal of slot filling is to collect from a corpus information regarding certain attributes ("slots") of an entity, which may be a person or some type of organization. Many search tasks suffer from two major problems; the challenges presented by query disambiguation and answer selection can impact the quality of results and the high-cost associated with human annotation of training data. The increasing number of diverse approaches based on different resources provide new opportunities to benefit from collaborative search (i.e. crowd-sourcing) by multiple human annotators or systems. Previous collaborative search methods had two underlying assumptions: (1) more sources are better; that is, novice annotators can be trained to perform annotation with reasonable quality; (2) diversity is beneficial in that human annotators tend to make different decisions, which can result in complementary information. In addition to describing how smarter systems can be used to expedite the creation of gold-standard answer keys, we discuss how the basic assumptions are not always valid and why it becomes crucial to pursue alternative resources instead of more human annotators.

SMER'11, October 28, 2011, Glasgow, Scotland, UK

1.1 System Combination Settings

We use NIST's Text Analytics Conference (TAC)'s Knowledge Base Population (KBP) track [9] as a case study. Many existing methodologies have been used to address this task and while each method has inherent strengths and weaknesses, the increasing number of system submissions, shared resources and diversity of systems have made it possible to benefit from system combination. Our work examines the extent to which different combination techniques can exceed the individual system limitations and demonstrate the effectiveness of a supervised re-ranking method for combining systems in the two experimental settings.

It has been shown that the combination of several algorithms for a prediction problem can result in better performance than any one system alone. In Table 1 we formalize the characteristics of baseline systems in different combination settings. The "perfect" setting is when a large number of systems are available, they are developed from sufficiently diverse resources and have comparable performance. However, in practice these conditions are not always satisfied and combination techniques must be robust to conditions that are less than ideal. Our work focuses on automatically combining slot filling systems in two settings: (1) *low transparency setting*; the baseline systems serve as "black boxes" to the combiner, providing minimal intermediate information for candidate answers; (2) *high transparency setting*; the baseline systems provide intermediate system results that can allow for enhanced feature encoding for the combiner. By studying features, rules and experimental conditions that are typically beneficial for re-ranking multiple systems, we conclude that the most effective re-ranking features can be categorized into general types. Compared with simple voting and merging based combination, we will show

Conditions	Perfect Setting	Common Settings	
		Low Transparency	High Transparency
Many baseline systems	Yes	Yes	No
Intermediate results available	Yes	No	Yes
Diverse algorithms	Yes	Yes	Yes
Diverse data	Yes	No	No
Diverse linguistic resources	Yes	Yes	No
Example	-	multiple system submissions in a shared task	different methods implemented by the same developer

Table 1: Characteristics of System Combination Settings

that supervised re-ranking approaches are most effective for system combination.

We use the slot filling task at the NIST Text Analytics Conference (TAC)'s Knowledge Base Population (KBP) track [9] as a case study. Many existing methodologies have been used to address this task. While each method has inherent strengths and weaknesses, the increasing number of system submissions, shared resources and diversity of systems have made it possible to benefit from system combination. Our work examines the extent to which different combination techniques can exceed the individual system limitations and demonstrate the effectiveness of a supervised re-ranking method for combining systems in the two common settings we discuss above.

Lastly, further advances in the slot filling task require a more effective way to generate complete answer keys at a reasonable cost to researchers. As a promising alternative to adding more humans for exhaustive annotation, we use a statistical re-ranking technique to combine the top systems from KBP2010, and demonstrate how this hybrid method can be used to construct high quality annotations more efficiently and at less cost than exhaustive human annotation.

2. RELATED WORK

In recent years, re-ranking techniques have been successfully applied to enhance the performance of NLP systems. A baseline generative model produces N-best hypotheses, which are then re-ranked using a rich set of local and global features in order to select the best hypothesis. There has been a considerable body of work on various trainable ranking algorithms such as Coordinate Descent RankBoost [13], PRank [6], Support Vector Machine-based Ranking (SVM-Rank) and MaxEnt Rank, with the most intensive study devoted to substantial improvements in name tagging [4, 10], parsing [2, 5, 7] and machine translation [8, 15]. Almost all of these methods focused on improving one stand-alone system. In contrast, we use a maximum entropy re-ranker to improve the quality of multiple slot filling systems in the low transparency setting where intermediate system results are unknown. We assess potential performance gains and attempt using a supervised re-ranking framework to achieve an efficient answer key generation method.

3. SUPERVISED RE-RANKING

In this section, we discuss the limitations of alternative system combination methods and our motivations for applying statistical re-ranking.

3.1 Voting, Union and System Performance

Merging individual systems to form the union of answers can be used as a simple system combination technique; however, this approach is not always ideal. The union guarantees that a higher recall can be achieved; however, when the differences in overall performance among systems is dramatic, the combined precision can be considerably lower than that of the best individual system. Also, union based methods present contradiction for single slots that basic merging does not address.

Voting provides an advantage over the union of results in that it employs a democratic approach to answer filtering. When compared with union, voting is less sensitive to individual system performance; however the method assumes that high frequency answers are more likely to be correct, which may not always be the case.

When the performance of the individual systems is known, another strategy to combine systems is performance-based. This combination is especially beneficial when the individual systems have good recall; however, in the case where the recall of individual systems is typically low, using a method that combines only the answers for the top system in the development can help to improve precision but may negatively impact recall.

3.2 Re-ranking Framework

Figure 1 depicts an overview of our new re-ranking framework and indicates two alternative processes to produce combined system output. The first represented by the solid arrows uses additional answer context provided by shared resources and representative of the *high transparency setting*. In this setting intermediate system results and pipeline specific answer confidence was available for feature encoding. The dotted line indicates the *low transparency setting* that is used when the re-ranker has no access to intermediate results, only the standard system output, and multiple systems function as black boxes.

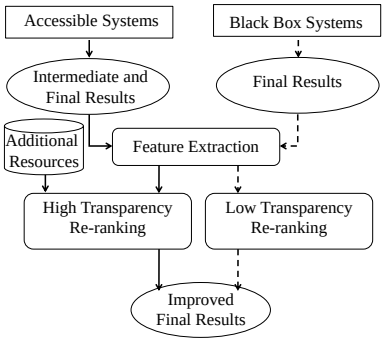

Figure 1: Supervised Re-ranking for System Combination Overview

3.3 Maximum Entropy Re-ranking

In this paper we choose maximum entropy (MaxEnt) models to train our re-rankers. For each answer *ans*, we use (1) to obtain an adjusted confidence, conf_{adj}, using the system's confidence score, conf_b, and the probability of the class la-

bel indicated my a maximum entropy classifier, conf_c. In the high transparency setting, baseline confidence values, conf_b, of individual systems can provide an estimate that is generally useful for system level answer validation; however, when combining confidence estimations across multiple systems metrics can be incongruent in terms of scale, reliability, and accuracy. Before re-ranking individual system confidence values are rescaled from 0 to 1. In the last step a threshold is used to filter the lowest confidence answers. For individual systems that provide no baseline confidence values (e.g. low transparency setting), one can still use MaxEnt re-ranking by assigning conf_b to a uniform value.

$$\text{conf}_{adj}(ans) = \text{conf}_b(ans) + \text{conf}_c(ans) \quad (1)$$

3.4 Re-ranking Feature Design

We identify global and new validation features to be defined across various applications in re-ranking. Global features across systems are these are features which capture the interactions across system outputs, thus they are not available in the individual system run time, such as the frequency of each candidate in the merged system output, or selecting the most confident answer for a slot-type that is single-valued. Such features can be aggregated across all systems and weighted by certain attributes (e.g. system IDs) in both low transparency and high transparency settings. In the high transparency setting, we can identify new validation features which are missed by all individual systems, and exploit additional resources to encode them. In addition, since we can access intermediate results, we can also analyze them (e.g. apply dependency parsing to the context sentences of all results) using new resources and encode new validation features.

4. CASE STUDY IN SLOT FILLING

The goal of slot filling is to collect from the corpus information regarding certain attributes ("slots") of an entity, which may be a person or some type of organization. Each query in the Slot Filling task consists of the name of the entity, its type (person or organization), a background document containing the name (again, to disambiguate the query in case there are multiple entities with the same name), its node ID (if the entity appears in the knowledge base), and the attributes which need not be filled. Attributes are excluded if they are already filled in the reference data base and can only take on a single value. Along with each slot fill, the system must provide the ID of a document which supports the correctness of this fill. If the corpus does not provide any information for a given attribute, the system should generate a NIL response (and no document ID). KBP2010 defined 26 types of attributes for persons (such as the age, birthplace, spouse, children, job title, and employing organization) and 16 types of attributes for organizations (such as the top employees, the founder, the year founded, the headquarters location, and subsidiaries). Some of these attributes are specified as only taking a single value (e.g., birthplace), while some can take multiple values (e.g., top employees).

The reference knowledge base (KB) includes several hundred thousand entities and is based on articles from an October 2008 dump of English Wikipedia which includes 818,741 entries. The source collection includes 1,286,609 newswire documents, 490,596 web documents and hundreds of transcribed spoken documents.

System	org:subsidiaries	per:age
sys A	0.0	0.0
sys B	20.0	81.5
sys C	66.7	56.5
sys D	-	76.9
sys E	-	13.8

Table 2: Precision (%) of 5 Systems on KBP Slots

To score slot filling, we first pool all system responses together with a set of manually prepared slot fills by Linguistic Data Consortium. These responses are then assessed by manual review. Each system response is rated as correct, wrong, inexact or redundant. Given these judgments, we calculate the precision, recall and F-measure of each system, as defined in [9]. To train the re-rankers, we used annotated system results from the 2009 and 2010 TAC KBP evaluation to learn the feature weights for the model.

4.1 Low Transparency Setting

In the low transparency setting, we conduct experiments on the top 3 systems in the KBP2009 evaluation and the top 13 systems in KBP2010 evaluation. Although for each KBP task several additional systems provided answers, we limited the experiment to only those systems with reasonable performance.

Table 2 shows the variable precision among top KBP2010 systems for two arbitrary systems and slots, *org:subsidiaries* and *per:age*. Of the top 13 systems, only 9 produced answers for the *org:subsidiaries* slot. The average precision was 20%, showing a range of 67 points. For the *per:age* slot, all 13 systems generated answers, reporting an average precision of 44.5%, and a range of 85 points. By examining system output by slot type, we again observe that within a system, the performance can be notably different across various slot types, and that variable precision exists among different systems for the same slot type. For example, three systems that did not produce any results for the subsidiaries slot, show well above average precision for *p*er:age, one at 28 points higher than the system average.

Table 3 describes the features used to re-rank systems when intermediate results are unknown. The number of features we can exploit for this setting is limited by a lack of additional answer information provided by a specific developer, or the unknown restrictions of individual systems. For example, some systems constructed patterns manually and so could not produce reliable confidence estimation; other systems used distant learning based answer validation and could not provide specific context sentences.

4.2 High Transparency Setting

In the high transparency setting, we applied a high performing slot filling system that consists of three pipelines using different algorithms: supervised information extraction (IE), pattern learning (PL) and question answering (QA) [3]. The supervised IE pipeline includes Automated Content Extraction (ACE)[1] relation extraction and event extraction based on MaxEnt models that incorporate diverse lexical, syntactic, semantic and ontological knowledge. The

[1]http://www.itl.nist.gov/iad/mig/tests/ace/

Feature Descriptions
System and Slot Type: identifies the system of origin and the slot type
Number of Tokens and Slot Type: the number of tokens in the answer by the slot type
Answer Frequency: for each *q,a* pair, the answer frequency it appears in the answer set for the same slot type

Table 3: Features for the Low Transparency Setting

Slot	IE	QA	PM
org:city_of_headquarters	**71.4**	55.0	-
org:country_of_headquarters	**75.0**	15.8	16.7

Table 4: Pipeline Precision (%) for Headquarters Slot Type

extracted relations and events are then mapped to KBP slot fills. In pattern learning, we extract and rank patterns based on a distant supervision approach [12] that uses entity attribute pairs from Wikipedia Infoboxes and Freebase [1]. Then, we apply these patterns to extract attributes for unseen entities. To further complement our final answer set, we also apply an open domain QA system, OpenEphyra [14] to retrieve additional candidate answers that may have not been identified by IE or pattern learning. To estimate the relevance of a query and answer pair obtained from the QA system in the KBP source collection, we use a corrected conditional probability based metric [11] for answer validation and filtering on the answer's name type.

We assess the contribution of high transparency re-ranking on the KBP2009 evaluation data set using our CUNY systems. Although our PL pipeline was the overall best performer among the three pipelines, slot-level performance was variable, and we aimed to integrate our individual pipelines to highlight the strengths and mitigate the weaknesses of the component systems. Not only is MaxEnt re-ranking theoretically more robust to the limiting conditions of union and voting methods, a well defined feature set has the potential to improve performance by eliminating spurious answers. For example, Table 4 shows that for the slot headquarters, IE pipeline has notably higher precision, but that some good answers are returned by PL and QA. Our re-ranking features aim to extract quality answers even when the overall performance of an individual system is poor by increasing the confidence of answers based on global features and decreasing the confidence of answers that show features generally associated with poor performance.

In the high transparency setting, access to intermediate results allowed for the encoding of additional features shown in Table 4.2. They were derived from either a context sentence, the dependency parse of the context sentence, a gazetteer, or a system specific confidence value.

5. EXPERIMENTAL RESULTS

To assess potential gains from system combination, we compare the performance of alternative combination methods including merging, voting and statistical re-ranking, based on five-folder cross validation with the alternative methods and relative to each individual system.

Categorical Feature Description
Answer Name Type: the name type of the candidate answer
Dependence Parse: for a system provided context sentence, the dependency parse is returned as a syntax tree
Binary Feature Description
Trigger Words: if a slot type related trigger word is in the system provided context sentence
Length of Dependence Parse: if the length of a context sentence's dependency parse is below a length threshold
Comma Delimited List: if the context sentence is a long comma delimited list
Query Subset of Answer: if the query is a subset of the answer
Invalid Answer: if an answer is listed in set of pre-defined invalid answers (e.g., "the" or "city")
Date Validation: for date slots, if an answer is in date format
Age Validation: for age slots, if an answer is in age range [0, 130]
Number Validation: for slots that should represent a quantity, if an answer was a real number
Country, City, Nationality and Title Validation: for each a specific slot, if the answer is found in an appropriate gazetteer

Table 5: Feature Descriptions for the High Transparency Setting

Although performance gains are not as striking, for system combination in a **low transparency setting**, re-ranking was shown to be beneficial, reporting 4.3% higher F-score than the best performing baseline system, and almost 2.7% higher that the union of results. This suggests that the performance is improved by using answer features in several important ways, including, but not limited to: identifying the strengths and weaknesses of individual systems to disproportionately weight answers based on answer features versus system level information, using answer features that are independent of an individual system to determine what global and slot specific characteristics are associated with "good" and "bad" answer quality, and mitigating errors attributed to the limitations of system specific metrics that are used to assess baseline confidence.

Table 6 compares the performance of re-ranking in **high transparency setting**. Given enough answer context, our results show that statistical re-ranking techniques has the greatest overall impact in terms of performance gains, and can exceed the performance of the top system (6.6% absolute improvement on F-measure). In contrast, the simple union and voting approaches did not provide significant gains.

Method	P	R	F
Feature-based IE	24.2	14.2	17.9
Pattern Matching	21.9	37.7	27.7
Question Answering	26.7	17.3	21.0
Voting	40.1	19.4	26.1
Union	18.3	53.5	27.3
Low Transparency	24.2	47.3	**32.0**
High Transparency	28.0	44.3	**34.3**

Table 6: Performance in High Transparency Setting(%)

In addition, we also evaluated the performance of using only low transparency features in the high transparency setting. The results demonstrated that the rich high transparency features about the answers and their contexts provided *significant of gains (2.3%), at a 98.4% confidence level* based on Wilcoxon Matched-Pairs Signed Ranks Test.

5.1 Impact of Important Features

In this section we investigate the detailed impact of important re-ranking features for slot filling and give specific examples to demonstrate their contributions.

System and Slot Type: We have shown that KBP systems report variable performance among systems and slots. Based on the empirical estimations of the MaxEnt model, this joint feature can be used to effectively modify slot level confidence considering system of origin. For example, the analysis of re-ranked output suggests this feature helps mitigate the impact of errors produced by systems that use co-occurrence based confidence values. For example, due to the high sentence level co-occurrence of *Dee Dee Meyers*, the former White House Press Secretary, with her former boss, *Bill Clinton*, a high answer confidence was erroneously assigned; however, based on poor performance associated with the slot *per:children* in the training set, after re-ranking the overall confidence was reduced below the filtering threshold.

Number of Tokens and Slot Type: In the low transparency setting "*1 X org:top_members/employees*" was a top feature, indicating a token count of 1 associated with the specific slot. Answers associated with this feature value had their confidence reduced, and many were eventually removed. To this end, re-ranking helped to filter inexact or incomplete answers that negatively impact overall system performance.

Voting: In general, answer frequency was not a top feature, but a significant one in the low transparency setting, where many systems are available. Answers with frequency of 1 did not bring dramatic changes to an answers confidence, but effected a large number or answers (40%) by reducing the answer confidence. Frequency as a feature in contrast to a stand-alone combination method has a major advantage – answers can be penalized for the presence of the feature, but not entirely eliminated. As noted earlier, removing all answers with a frequency of 1 can results in the elimination of many correct answers, especially when systems use diverse resources.

6. RE-RANKING FOR ANSWER-KEY CREATION

To facilitate the development of state-of-the-art slot filling systems, there is a current need for better methods to generate complete answer keys. We will describe the current method of answer key generation and discuss the potential of various combinations across automated systems to achieve higher quality annotations at a lower cost.

6.1 Human Annotation Bottleneck

Since it requires the annotators to conduct an exhaustive search in a large scale corpus, slot filling annotation is a costly and laborious task. Figure 2 shows the number of correct answers for 3 persons and 3 organizations after merging the results from multiple human annotators. From this figure, we can observe that a single human annotator can only find 70% of the answers merged from five annotators. Alternatively, the annotation set tends to converge by the time the fifth annotator is added. That is, the fifth annotator only found 8 new answers compared to the set of 342 answers merged previously. The precision, recall and F-measure for LDC human annotation - scored against the final answer key consisted of manual assessment on pooled answers from systems and human annotators - are only 70.14%, 54.06% and 61.06% respectively.

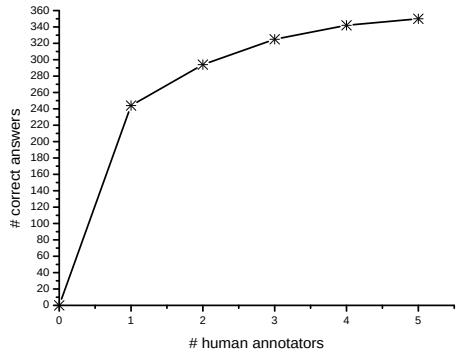

Figure 2: Community Effort - Human Annotation Combination

6.2 Assessment Methods Comparison

In this section we will discuss the possibilities of various combinations including our statistical re-ranking approach to produce annotated data, either using direct manual annotation or by the assessment of system output. For the comparison, we used the top 13 system output in KBP2010 evaluation with different assessment methods as shown in Table 7.

	Method	Description
1	Baseline	Alphabetical order (by site name)
2	Voting	Majority voting; the responses which get more votes across systems are assessed first
3	Re-ranking	Re-ranked by our statistical re-ranker
4	Oracle 1	Ranked by the oracle system performance (the best to the worst)
5	Oracle 2	Assess all correct responses first

Table 7: System Combination Methods for Automatic Assessment

Figure 3 summarizes the results of these methods. The common end point of curves 1-5 represents the cost and benefit of assessing all system responses.

Comparing curve 5 (pure manual annotation) with points along curves 1 to 4 allows us to compare manual annotation with various strategies for assessing pooled system responses. The baseline (curve 1) is less efficient than direct manual annotation. This reflects the fact that some of the systems have very low precision. However, if we employ our statistical re-ranking approach (curve 2) and apply some cut-off, the process (109.6 person hours) can be dramatically more efficient than manual annotation (300 person hours) and maintain comparable recall. In particular, the performance of the statistical re-ranking approach is very close to Oracle 1. This suggests an alternative way to annotate answer keys for slot filling is to re-rank and assess the pooled system responses as opposed to identifying correct answers from scratch.

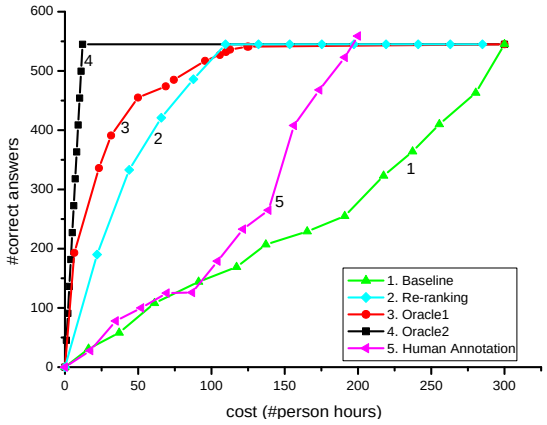

Figure 3: Answer-Key Creation Method Comparison

7. CONCLUSIONS AND FUTURE WORK

The KBP task is a new and active area of study and a variety of methodologies have been applied, providing opportunities for system combination. We describe a novel system combination approach based on supervised re-ranking and categorize the framework according using different combination settings and feature categories; thus one only needs to materialize the general features and settings to apply our approach to a new research problem.

Furthermore, we investigated how combinations of automatic systems have the potential to assist in the generation of answer keys and discuss the limitations of crowdsourcing in the context of KBP. Our findings suggest that adding automated systems with reasonable performance that used new resources instead of additional human annotators can expedite the generation of more complete answer keys – a labor intensive and cost prohibitive process when performed by human annotators alone. The techniques we have developed and the lessons we have learned may prove equally useful in other domains where human annotation alone fails to capture the prohibitively large space of results in a data set.

From the results we also observed that the learning curve converges quickly, indicating some low ranked systems did not contribute to the generation of additional correct answers. On the other hand, poor performing systems tended to produce an overwhelming amount of incorrect answers, stressing the importance of automatically detecting the stopping criteria as our future work, in order to save human assessment time.

8. ACKNOWLEDGEMENTS

This work was supported by the U.S. Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NS-CTA), the U.S. NSF CAREER Award under Grant IIS-0953149, the U.S. NSF EAGER grant No. 1144111 and PSC-CUNY Research Program. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

9. REFERENCES

[1] Bollacker, K., Cook, R., and Tufts, P. Freebase: A shared database of structured general human knowledge. In *Proc. National Conference on Artificial Intelligence* (2008).

[2] Charniak, E., and Johnson, M. Coarse-to-fine N-best parsing and MaxEnt discriminative reranking. In *Proc. ACL2005* (2005).

[3] Chen, Z., Tamang, S., Lee, A., Li, X., Passantino, M., and Ji, H. Top-down and bottom-up: A combined approach to slot filling. *Lecture Notes in Computer Science 6458* (December 2010), 300–309.

[4] Collins, M., and Duffy, N. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proc. ACL2002* (2002).

[5] Collins, M., and Koo, T. Discriminative reranking for natural language parsing. In *Journal of Association for Computational Linguistics* (2003).

[6] Crammer, K., and Singeri, Y. PRanking with ranking. In *Proc. NIPS2001* (2001).

[7] Henderson, J., and Titov, I. Data-defined kernels for parse reranking derived from probabilistic models. In *Proc. ACL2005* (2005).

[8] Huang, F., and Papeneni, K. Hierarchical system combination for machine translation. In *Proc. EMNLP2007* (2007).

[9] Ji, H., Grishman, R., Dang, H. T., Griffitt, K., and Ellis, J. An overview of the TAC2010 knowledge base population track. In *Proc. TAC 2010* (2010).

[10] Ji, H., Rudin, C., and Grishman, R. Re-ranking algorithms for name tagging. In *Proc. HLT/NAACL 06 Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing* (2006).

[11] Magnini, B., Negri, M., Prevete, R., and Tanev, H. Is it the right answer? exploiting web redundancy for answer validation. In *Proc. ACL2002* (2002).

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

[13] Rudin, C., Cortes, C., Mohri, M., and Schapire, R. E. Margin-based ranking and boosting meet in the middle. In *Proc. the 18th Annual Conference on Learning Theory, year = 2005,*.

[14] Schlaefer, N., Ko, J., Betteridge, J., Sautter, G., Pathak, M., and Nyberg, E. Semantic extensions of the Ephyra QA system for TREC 2007. In *Proc. TREC 2007* (2007).

[15] Shen, L., Sarkar, A., and Och, F. J. Discriminative reranking for machine translation. In *Proc. NAACL 2004* (2004).