To Appear In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP-16) 2016.

Combining Supervised and Unsupervised Ensembles for Knowledge Base Population

Nazneen Fatema Rajani and **Raymond J. Mooney**
Department Of Computer Science
The University of Texas at Austin
nrajani@cs.utexas.edu, mooney@cs.utexas.edu

Abstract

We propose an algorithm that combines supervised and unsupervised methods to ensemble multiple systems for two popular Knowledge Base Population (KBP) tasks, Cold Start Slot Filling (CSSF) and Tri-lingual Entity Discovery and Linking (TEDL). We demonstrate that it outperforms the best system for both tasks in the 2015 competition, several ensembling baselines, as well as a state-of-the-art stacking approach. The success of our technique on two different and challenging problems demonstrates the power and generality of our combined approach to ensembling.

1 Introduction

Ensembling multiple systems is a well known standard approach to improving accuracy in several machine learning applications (Dietterich, 2000). Ensembles have been applied to parsing (Henderson and Brill, 1999), word sense disambiguation (Pedersen, 2000), sentiment analysis (Whitehead and Yaeger, 2010) and information extraction (IE) (Florian et al., 2003; McClosky et al., 2012). Recently, using *stacking* (Wolpert, 1992) to ensemble systems was shown to give state-of-the-art results on slot-filling and entity linking for *Knowledge Base Population* (KBP) (Viswanathan et al., 2015; Rajani and Mooney, 2016). Stacking uses supervised learning to train a meta-classifier to combine multiple system outputs; therefore, it requires historical data on the performance of each system. Rajani and Mooney (2016) use data from the 2014 iteration of the KBP competition for training and then test on the data from the 2015 competition, therefore they can only ensemble the *shared systems* that participated in both years.

However, we would sometimes like to ensemble systems for which we have no historical performance data. For example, due to privacy, some companies may be unwilling to share their performance on arbitrary training sets. Simple methods such as voting permit "unsupervised" ensembling, and several more sophisticated methods have also been developed for this scenario (Wang et al., 2013). However, such methods fail to exploit supervision for those systems for which we *do* have training data. Therefore, we present an approach that utilizes supervised *and* unsupervised ensembling to exploit the advantages of both. We first use unsupervised ensembling to combine systems without training data, and then use stacking to combine this ensembled system with other systems with available training data.

Using this new approach, we demonstrate new state-of-the-art results on two NIST KBP challenge tasks: *Cold Start Slot-Filling* (CSSF)[1] and the *Trilingual Entity Discovery and Linking* (TEDL) (Ji et al., 2015). Our approach outperforms the best system as well as other state-of-the-art ensembling methods on both tasks in the most recent 2015 competition. There is one previous work on ensembling supervised and unsupervised models using graph-based consensus maximization (Gao et al., 2009), however we show that it does not do as well as our stacking method.

[1] http://www.nist.gov/tac/2015/KBP/ColdStart/guidelines.html

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1943–1948,
Austin, Texas, November 1-5, 2016.

2 Overview of KBP Tasks

2.1 Cold Start Slot Filling (CSSF)

The goal of CSSF is to collect information (fills) about specific attributes (slots) for a set of entities (queries) from a given corpus. The query entities can be a person, organization, or geo-political entity (PER/ORG/GPE). The input is a set of queries along with a text corpus in which to look for information. The output is a set of slot fills for each query. Systems must also provide *provenance* in the form of *docid:startoffset-endoffset*, where *docid* specifies a source document and the offsets demarcate the text in this document supporting the filler. Systems may also provide a confidence score to indicate their certainty in the extracted information.

2.2 Tri-lingual Entity Discovery and Linking (TEDL)

The first step in the TEDL task is to discover all entity mentions in a corpus with English, Spanish and Chinese documents. The entities can be a person, organization or geo-political entity (PER/ORG/GPE) and in 2015 two more entity types were introduced – facility and location (FAC/LOC). The extracted mentions are then linked to an existing English KB (a version of FreeBase) entity via its ID. If there is no KB entry for an entity, systems are expected to cluster all the mentions for that entity using a NIL ID. The output for the task is a set of extracted mentions, each with a string, its provenance in the corpus, and a corresponding KB ID if the system could successfully link the mention, or else a mention cluster with a NIL ID. Systems can also provide a confidence score for each mention.

3 Ensembling Algorithm

Figure 1 illustrates our system which trains a final meta-classifier for combining multiple systems using confidence scores and other auxiliary features depending on the task.

3.1 Supervised Ensembling Approach

For the KBP systems that are common between years, we use the stacking method of Viswanathan et al. (2015) for these shared systems.

The meta-classifier makes a binary decision for each distinct output represented as a *key-value* pair.

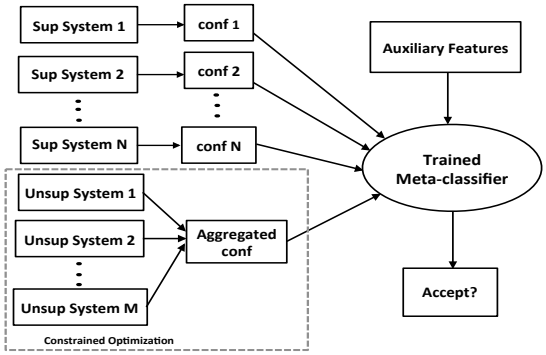

Figure 1: Illustration of our approach to combine supervised and unsupervised ensembles.

The function of the *key* is to provide a handle for aggregating outputs that are common across systems. For the CSSF task, the *key* for ensembling multiple systems is a query along with a slot type, for example, per:age of "Barack Obama" and the *value* is a computed *slot fill*. For TEDL, the key is the *KB (or NIL) ID* and the value is a *mention*, that is a specific reference to an entity in the text. The top half of Figure 1 illustrates ensembling multiple systems with historical training data using a supervised approach.

3.2 Unsupervised Ensembling Approach

Only 38 of the 70 systems that participated in CSSF 2015 also participated in 2014, and only 24 of the 34 systems that participated in TEDL 2015 also participated in 2014 EDL. Therefore, many KBP systems in 2015 were new and did not have past training data. In fact, some of the new systems performed better than the shared systems, for example the *hltcoe* system did not participate in 2014 but was ranked 4^{th} in the 2015 TEDL task (Ji et al., 2015). Thus, for improving *recall* and performance in general, it is crucial to use systems without historical training data, which we call unsupervised systems. To achieve this end, we first ensemble such systems using an unsupervised technique. Frequently, the confidence scores provided by systems are not well-calibrated probabilities. So in order to calibrate the confidence scores across unsupervised systems, we use the constrained optimization approach proposed by Wang et al. (2013). The idea is to aggregate the raw confidence values produced by individual KBP systems,

to arrive at a single aggregated confidence value for each *key-value* pair. The constraints ensure that the aggregated confidence score is close to the raw score as well as proportional to the agreement among systems on a value for a given key. Thus for a given key, if a system's value is also produced by multiple other systems, it would have a higher score than if it were not produced by any other system. The authors use the inverse ranking of the average precision previously achieved by individual systems as the weights in their algorithm. However since we use this approach for systems with no historical performance data, we use uniform weights across all unsupervised systems for both the tasks.

We use the slot type for the CSSF task and entity type for the TEDL task to define the constraints on the values. The output from the constrained optimization approach for both tasks is a set of key-values with aggregated confidence scores across all unsupervised systems which go directly into the stacker as shown in the lower half of Figure 1. Using the aggregation approach as opposed to directly using the raw confidence scores allows the classifier to meaningfully compare confidence scores across multiple systems although they are produced by very diverse systems.

Another unsupervised ensembling method we tried in place of the constrained optimization approach is the Bipartite Graph based Consensus Maximization (BGCM) approach of Gao et al. (2009). BGCM is presented as a way of combining supervised and unsupervised models, so we compare it to our stacking approach to combining supervised and unsupervised systems, as well as an alternative approach to ensembling *just* the unsupervised systems before passing their output to the stacker. BGCM performs an optimization over a bipartite graph of systems and outputs, where the objective function favors the smoothness of the label assignments over the graph, as well as penalizing deviations from the initial labeling provided by supervised models.

3.3 Combining Supervised and Unsupervised

We propose a novel approach to combine the aforementioned supervised and unsupervised methods using a stacked meta-classifier as the final arbiter for accepting a given key-value. The outputs from the supervised and unsupervised systems are fed into the stacker in a consistent format such that there is a unique input *key-value* pair. Most KBP teams submit multiple variations of their system. Before ensembling, we first combine multiple runs of the same team into one. Of the 38 CSSF systems from 10 teams for which we have 2014 data for training and the 32 systems from 13 teams that do not have training data, we combine the runs of each team into one to ensure diversity of the final ensemble. For the slot fills that were common between the runs of a given team, we compute an average confidence value, and then add any additional fills that are not common between runs. Thus, we obtained 10 systems (one for each team) for which we have supervised data for training stacking. Similarly, we combine the 24 TEDL systems from 6 teams that have 2014 training data and 10 systems from 4 teams that did not have training data into one per team. Thus using the notation in Figure 1, for TEDL, $N = 6$ and $M = 4$ while for CSSF, $N = 10$ and $M = 13$.

The unsupervised method produces aggregated, calibrated confidence scores which go directly into our final meta-classifier. We treat this combination as a single system called the *unsupervised ensemble*. We add the unsupervised ensemble as an additional system to the stacker, thus giving us a total of $N+1$, that is 11 CSSF and 7 TEDL systems. Once we have extracted the auxiliary features for each of the N supervised systems and the unsupervised ensemble for both years, we train the stacker on 2014 systems, and test on the 2015 systems. The unsupervised ensemble for each year is composed of different systems, but hopefully the stacker learns to combine a generic unsupervised ensemble with the supervised systems that are shared across years. This allows the stacker to arbitrate the final correctness of a key-value pair, combining systems for which we have no historical data with systems for which training data *is* available. To learn the meta-classifier, we use an L1-regularized SVM with a linear kernel (Fan et al., 2008) (other classifiers gave similar results).

3.4 Post-processing

Once we obtain the decisions on each key-value pair from the stacker, we perform some final post-processing. For CSSF, each list-valued slot fill that is classified as correct is included in the final output. For single-valued slot fills, if they are multiple fills

Methodology	Precision	Recall	F1
Combined stacking and constrained optimization with auxiliary features	0.4679	**0.4314**	**0.4489**
Top ranked SFV system in 2015 (Rodriguez et al., 2015)	0.4930	0.3910	0.4361
Stacking using BGCM instead of constrained optimization	**0.5901**	0.3021	0.3996
BGCM for combining supervised and unsupervised systems	0.4902	0.3363	0.3989
Stacking with auxiliary features described in (Rajani and Mooney, 2016)	0.4656	0.3312	0.3871
Ensembling approach described in (Viswanathan et al., 2015)	0.5084	0.2855	0.3657
Top ranked CSSF system in 2015 (Angeli et al., 2015)	0.3989	0.3058	0.3462
Oracle Voting baseline (3 or more systems must agree)	0.4384	0.2720	0.3357
Constrained optimization approach described in (Wang et al., 2013)	0.1712	0.3998	0.2397

Table 1: Results on 2015 Cold Start Slot Filling (CSSF) task using the official NIST scorer

Methodology	Precision	Recall	F1
Combined stacking and constrained optimization	0.686	**0.624**	**0.653**
Stacking using BGCM instead of constrained optimization	0.803	0.525	0.635
BGCM for combining supervised and unsupervised outputs	0.810	0.517	0.631
Stacking with auxiliary features described in (Rajani and Mooney, 2016)	0.813	0.515	0.630
Ensembling approach described in (Viswanathan et al., 2015)	**0.814**	0.508	0.625
Top ranked TEDL system in 2015 (Sil et al., 2015)	0.693	0.547	0.611
Oracle Voting baseline (4 or more systems must agree)	0.514	0.601	0.554
Constrained optimization approach	0.445	0.176	0.252

Table 2: Results on 2015 Tri-lingual Entity Discovery and Linking (TEDL) using official NIST scorer and CEAF metric

that were classified as correct for the same query and slot type, we include the fill with the highest meta-classifier confidence.

For TEDL, for each entity mention link that is classified as correct, if the link is a KB cluster ID then we include it in the final output, but if the link is a NIL cluster ID then we keep it aside until all mention links are processed. Thereafter, we resolve the NIL IDs across systems since NIL ID's for each system are unique. We merge NIL clusters across systems into one if there is at least one common entity mention among them.

4 Experimental Results

All results were obtained using the official NIST scorers after the competitions ended.[2] We compare to the purely supervised approach of Viswanathan et al. (2015) using shared systems between 2014 and 2015, and the constrained optimization approach of Wang et al. (2013) using all 2015 systems. We also compare to BGCM (Gao et al., 2009) in two ways. First, we use BGCM in place of the constrained optimization approach to ensemble unsupervised systems while keeping the rest of our pipeline the same. Secondly, we also compare to combining both supervised and unsupervised systems using BGCM instead of stacking. We also include an *"oracle"* voting ensembling baseline, which varies the threshold on the number of systems that must agree to identify an "oracle" threshold that results in the highest F1 score for 2015. We find that for CSSF a threshold of 3, and for TEDL a threshold of 4, gives us the best F1 score.

Tables 1 and 2 show CSSF and TEDL results. Our full system, which combines supervised and unsupervised ensembling performed the best on both tasks. TAC-KBP also includes the Slot Filler Validation (SFV) task[3] where the goal is to ensemble/filter outputs from multiple slot filling systems. The top ranked system in 2015 (Rodriguez et al., 2015) does substantially better than many of the other ensembling approaches, but it does not do as well as our best performing system. The purely

[2] http://www.nist.gov/tac/2015/KBP/ColdStart/tools.html, https://github.com/wikilinks/neleval

[3] http://www.nist.gov/tac/2015/KBP/SFValidation/index.html

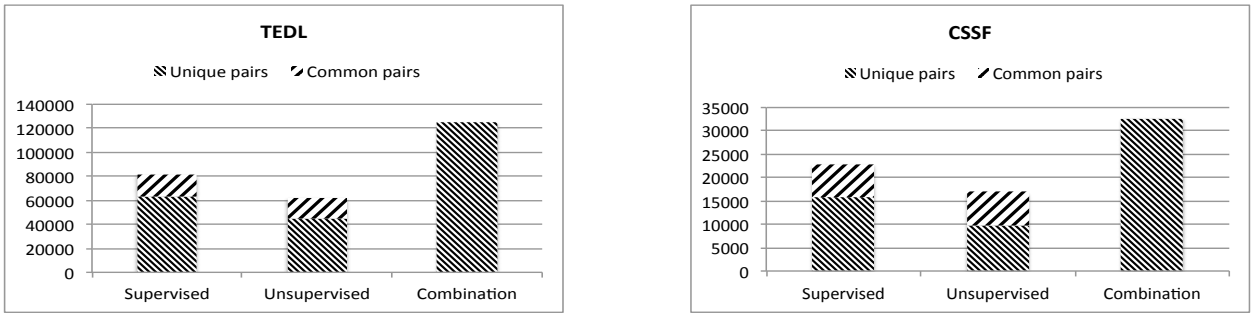

Figure 2: Total number of unique and common input pairs contributed by the supervised and unsupervised systems to the combination for the TEDL and CSSF tasks respectively.

supervised approach of Viswanathan et al. (2015) and the auxiliary features approach of Rajani and Mooney (2016) performs substantially worse, although still outperforming the top-ranked individual system in the 2015 competition. These approaches only use the common systems from 2014, thus ignoring approximately half of the systems. The approach of Wang et al. (2013) performs very poorly by itself; but when combined with stacking gives a boost to recall and thus the overall $F1$. Note that all our combined methods have a substantially higher recall. The oracle voting baseline also performs very poorly indicating that naive ensembling is not advantageous.

TEDL provides three different approaches to measuring accuracy: entity discovery, entity linking, and mention CEAF (Ji et al., 2015). CEAF finds the optimal alignment between system and gold standard clusters, then evaluates precision and recall micro-averaged. We obtained similar results on all three metrics and only include CEAF. The purely supervised stacking approach over shared systems does not do as well as any of our combined approaches even though it beats the best performing system (i.e. IBM) in the 2015 competition (Sil et al., 2015). The relative ranking of the approaches is similar to those obtained for CSSF, proving that our approach is very general and improves performance on two quite different and challenging problems.

Even though it is obvious that the boost in our recall was because of adding the unsupervised systems, it isn't clear how many new *key-value* pairs were generated by these systems. We thus evaluated the contribution of the systems ensembled using the supervised approach and those ensembled using the unsupervised approach, to the final combination for both the tasks. Figure 2 shows the number of unique as well as common *key-value* pairs that were contributed by each of the approaches. The unique pairs are those that were produced by one approach but not the other and the common pairs are those that were produced by both approaches. The number of unique pairs in the combination is the union of unique pairs in the supervised and unsupervised approaches. We found that approximately one third of the input pairs in the combination came from the unique pairs produced just by the unsupervised systems for both the TEDL and CSSF tasks. Only about 15% and 22% of the total input pairs were common between the two approaches for the TEDL and CSSF tasks respectively. Our findings highlight the importance of utilizing systems that do not have historical training data.

5 Conclusion

We presented results on two diverse KBP tasks, showing that a novel stacking-based approach to ensembling both supervised and unsupervised systems is very promising. The approach outperforms the top ranked systems from both 2015 competitions as well as several other ensembling methods, achieving a new state-of-the-art for both of these important, challenging tasks. We found that adding the unsupervised ensemble along with the shared systems specifically increased recall substantially.

Acknowledgment

This research was supported by the DARPA DEFT program under AFRL grant FA8750-13-2-0026.

References

Gabor Angeli, Victor Zhong, Danqi Chen, Arun Chaganty, Jason Bolton, Melvin Johnson Premkumar, Panupong Pasupat, Sonal Gupta, and Christopher D. Manning. 2015. Bootstrapped Self Training for Knowledge Base Population. In *Proceedings of the Eighth Text Analysis Conference (TAC2015)*.

T. Dietterich. 2000. Ensemble Methods in Machine Learning. In J. Kittler and F. Roli, editors, *First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. Named entity recognition through classifier combination. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 168–171. Association for Computational Linguistics.

Jing Gao, Feng Liang, Wei Fan, Yizhou Sun, and Jiawei Han. 2009. Graph-based consensus maximization among multiple supervised and unsupervised models. In *Advances in Neural Information Processing Systems (NIPS2009)*, pages 585–593.

John C. Henderson and Eric Brill. 1999. Exploiting Diversity in Natural Language Processing: Combining Parsers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP99)*, pages 187–194, College Park, MD.

Heng Ji, Joel Nothman, Ben Hachey, and Radu Florian. 2015. Overview of TAC-KBP2015 Tri-lingual Entity Discovery and Linking. In *Proceedings of the Eighth Text Analysis Conference (TAC2015)*.

David McClosky, Sebastian Riedel, Mihai Surdeanu, Andrew McCallum, and Christopher D Manning. 2012. Combining Joint Models for Biomedical Event Extraction. *BMC Bioinformatics*.

Ted Pedersen. 2000. A Simple Approach to Building Ensembles of Naive Bayesian Classifiers for Word Sense Disambiguation. In *North American Chapter of the Association for Computational Linguistics (NAACL2000)*, pages 63–69.

Nazneen Fatema Rajani and Raymond J. Mooney. 2016. Stacking With Auxiliary Features. *ArXiv e-prints*.

Miguel Rodriguez, Sean Goldberg, and Daisy Zhe Wang. 2015. University of Florida DSR lab system for KBP slot filler validation 2015. In *Proceedings of the Eighth Text Analysis Conference (TAC2015)*.

Avirup Sil, Georgiana Dinu, and Radu Florian. 2015. The IBM systems for trilingual entity discovery and linking at TAC 2015. In *Proceedings of the Eighth Text Analysis Conference (TAC2015)*.

Vidhoon Viswanathan, Nazneen Fatema Rajani, Yinon Bentor, and Raymond J. Mooney. 2015. Stacked Ensembles of Information Extractors for Knowledge-Base Population. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL2015)*, pages 177–187, Beijing, China, July.

I-Jeng Wang, Edwina Liu, Cash Costello, and Christine Piatko. 2013. JHUAPL TAC-KBP2013 Slot Filler Validation System. In *Proceedings of the Sixth Text Analysis Conference (TAC2013)*.

Matthew Whitehead and Larry Yaeger. 2010. Sentiment mining using ensemble classification models. In Tarek Sobh, editor, *Innovations and Advances in Computer Sciences and Engineering*. SPRINGER, Berlin.

David H. Wolpert. 1992. Stacked generalization. *Neural Networks*, 5:241–259.