Stacking with Auxiliary Features for Entity Linking in the Medical Domain

Nazneen Fatema Rajani
Department of Computer Science
University of Texas at Austin
nrajani@cs.utexas.edu

Mihaela Bornea **Ken Barker**
T.J. Watson Research Center
IBM Research
{mabornea, kjbarker}@us.ibm.com

Abstract

Linking spans of natural language text to concepts in a structured source is an important task for many problems. It allows intelligent systems to leverage rich knowledge available in those sources (such as concept properties and relations) to enhance the semantics of the mentions of these concepts in text. In the medical domain, it is common to link text spans to medical concepts in large, curated knowledge repositories such as the *Unified Medical Language System*. Different approaches have different strengths: some are precision-oriented, some recall-oriented; some better at considering context but more prone to hallucination. The variety of techniques suggests that *ensembling* could outperform component technologies at this task. In this paper, we describe our process for building a *Stacking* ensemble using additional, auxiliary features for Entity Linking in the medical domain. Our best model beats several baselines and produces state-of-the-art results on several medical datasets.

1 Introduction

Entity Linking is the task of mapping phrases in text (*mention spans*) to concepts in a structured source, such as a knowledge base. The mention span is usually a word or short phrase describing a single, coherent concept. For example, "back pain" may be a mention span for a *Dorsalgia* concept in a knowledge base. The *span context* is a window of text surrounding the mention span that may be useful for disambiguating it. For example, the sentence "The patient reports suffering from back pain for several years prior to treatment" may be useful for determining that "back pain" refers to the concept *Chronic Dorsalgia* in this context. In the medical domain, it is common to map mention spans to concepts in the *Unified Medical Language System* (*UMLS*)[1]. Concepts in UMLS have unique identifiers called *CUIs* (Concept Unique Identifiers). For example, the CUI for the concept *Dorsalgia* is C0004604.

The concepts in UMLS come from merging concepts from many disparate contributing vocabularies. Since automatic merging is imperfect, *UMLS* often contains multiple distinct CUIs for what amounts to the same semantic concept. For example, the three distinct CUIs C0425687, C1167958 and C3263244 are all *Jugular Venous Distension*. An Entity Linking system attempting to link a span such as "engorgement of the jugular vein" should be required to return all three CUIs. A ground truth dataset should include all the three mappings as well. *UMLS* also contains multiple textual labels for each CUI (called "variants") and semantic relations between CUIs, such as *Acetaminophen may_treat: Pain*.

Ensembling multiple systems is a well known standard approach to improving accuracy in machine learning (Dietterich, 2000). Ensembles have been applied to a wide variety of problems in all domains of artificial intelligence including natural language processing (NLP). However, these techniques do not learn to discriminate adequately across the component systems and thus are unable to integrate them optimally. Combining systems intelligently is crucial for improving the overall performance. In this paper, we use an approach called Stacking with Auxiliary Features (SWAF) (Rajani and Mooney, 2017) for combining multiple diverse models. Stacking (Wolpert, 1992) uses supervised learning to train a meta-classifier to

[1]UMLS: http://www.nlm.nih.gov/research/umls/

combine multiple system outputs. SWAF enables the stacker to fuse additional relevant knowledge from multiple systems and thus leverage them to improve prediction. The idea behind using auxiliary features is that an output is more reliable if not just multiple systems produce it but also agree on its provenance and there is sufficient supporting evidence. We are the first to use ensembling for entity linking in the medical domain that lacks labeled data. All the publicly available datasets are very small and thus learning is a problem. Our approach is designed to overcome these challenges in the medical domain by using auxiliary features that are precision-focused and can be used to form a classification boundary from small amounts of data.

2 Component Entity Linking Systems

The entity linking ensemble we have built includes eight component systems. Given a span of text, each component links the entities in text to zero or more matching concepts in UMLS. The ensemble examines all concepts produced by each component system for the given span and determines the final entity linking outcome. All the component systems use traditional rule-based methods and thus only perform well on certain types of concepts. The errors produced by these base systems are de-correlated and our goal is to leverage the systems to the fullest by using carefully designed auxiliary features. We used the following component systems in our ensemble.

Medical Concept Resolution: Three of the components systems are variations of the Medical Concept Resolution (MCR) approach introduced in (Aggarwal et al., 2015). The MCR systems find UMLS concepts that best capture the meaning of the input span as expressed in the textual context where the span appears. The algorithms consist of two main steps: candidate overgeneration and candidate ranking. Candidate overgeneration finds all concepts having any variant containing any of the tokens in the mention text. This step results in a large number of candidate concepts, many of them irrelevant. In the second step, the candidate concepts are ranked by measuring the similarity between mention context and candidate context. The mention context is a window of text surrounding the span. The candidate context is generated differently by each of the three MCR systems. Both the span context and the candidate context are treated as IDF-weighted bags-of-words for computing their cosine similarity. The higher the cosine similarity, the higher the rank of the candidate concept for the given span. The three variations of the MCR systems used are:

- *Gloss-Based MCR (GBMCR)*: generates the candidate context from the concept definitions in UMLS. In GBMCR, candidates are ranked according to the similarity between the words in the span mention (and its context) and the words in the UMLS definitions of the candidate.
- *Neighbor-Based MCR (NBMCR)*: generates the candidate context from the set of variants of the candidate's neighbors in UMLS. Neighbors are CUIs related to the candiate CUI by any of a select set of UMLS semantic relations. In NBMCR, candidates are ranked according to the similarity between the words in the span+context and the words in the variants of the candidate's neighbors.
- *Variants-Based MCR (VBMCR)*: generates the candidate context from the candidate's variants in UMLS. In VBMCR, candidates are ranked according to the similarity between the words in the span+context and the words in the candidate's variants.

Concept Mapper: Apache Concept Mapper matches text to dictionary entries. The dictionary contains surface forms and the concept identifiers those surface forms map to. The system included in the ensemble is based on a dictionary derived from the complete set of UMLS variants. Preprocessing of UMLS variants removes some superfluous acronyms (e.g. "nos" = "not otherwise specified"; "nec" = "not elsewhere classified"). The dictionary is also expanded beyond the UMLS variants by including adjective-to-noun and plural-to-singular transformations, as well as additional spelling variants and synonymous phrases derived from wikipedia redirect pages.

CUI Finder Verbatim (CFV): CFV (Aggarwal et al., 2015) is a dictionary-based system similar to ConceptMapper with advanced matching algorithms and synonym expansion. If no concept is found when matching the dictionary using the entire span, CFV attempts to find concepts for smaller windows by removing words from the span iteratively. The algorithm considers both left-to-right and right-to-left shrinking of the span. If

no concepts are found, it reduces the window size further. As soon as any concept is found, the algorithm stops, returning all concepts found for subspans of the given window size at any position within the original span.

MetaMap: This system is provided by the National Library of Medicine for detecting UMLS concepts in medical text.[2] It is NLP-based and uses domain-specific knowledge to map text to concepts. The ensemble includes MetaMap configured with the default settings.

cTAKES: Apache cTAKES[3] is an open source entity recognition system, originally developed at Mayo Clinic for identifying UMLS concepts in electronic medical records. cTAKES implements a terminology-agnostic dictionary lookup algorithm. Through the dictionary lookup, each named entity is mapped to a concept from the terminology. The dictionary lookup includes permutation of words in the spans, exact matches of the span and canonical forms of the words.

Structured Term Recognizer (STR): This system takes a span of text as input and produces a list of possible UMLS concepts for that span, as well as semantic types, if desired. Concept recognition proceeds in two phases: UMLS candidate generation and scoring of the candidate concepts. The candidate UMLS concepts are found by an inverted index, mapping tokens in the concepts to the concepts themselves. Once the candidate UMLS concepts are found, they are scored for similarity with the input span based on shared tokens and shared stems.

3 Stacking With Auxiliary Features

In this section we describe our algorithm and the auxiliary features used for classification. Figure 1 shows an overview of our ensembling approach.

3.1 Stacking

Stacking uses a meta-classifier to combine the outputs of multiple underlying systems. The stacker learns a classification boundary based on the confidence scores provided by individual systems for each possible output. Stacking has been shown to improve performance on tasks such as slot filling and tri-lingual entity linking (Viswanathan et al., 2015; Rajani and Mooney, 2016).

[2]MetaMap: http://metamap.nlm.nih.gov/
[3]cTAKES: https://ctakes.apache.org/

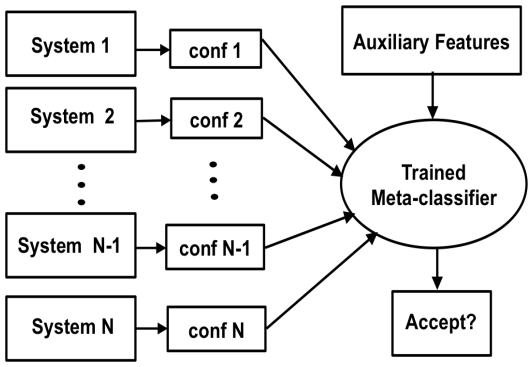

Figure 1: Ensemble Architecture using Stacking with Auxiliary Features. Given an input span, the ensemble judges every possible concept produced by the component systems and determines the final entity linking output.

3.2 Auxiliary Features

Stacking relies on systems producing a confidence score for every output. However, many times systems do not produce confidence scores or the scores produced are not probabilities or well calibrated and cannot be meaningfully compared. In such circumstances, it is beneficial to have other reliable auxiliary features. Auxiliary features enable the stacker to learn to rely on systems that not just agree on an output but also the provenance or the source of the output and other supporting evidence. We used four types of auxiliary features as part of our ensembling approach, described below.

3.2.1 CUI type

Every CUI in UMLS is associated with one or more semantic types (out of roughly 130 types). For example, the types associated with the CUI `C0000970` (acetaminophen) are `T109` (Organic chemical) and `T121` (Pharmacologic substance).

The CUI type is represented by a binary vector of size 130. The CUI type vector has ones for each associated semantic type of the CUI under consideration and zeros elsewhere. This CUI type vector is used as an auxiliary feature for ensembling. The CUI type enables the stacker to learn to rely on systems that perform better for certain CUI types.

3.2.2 Span-CUI document similarity

The second auxiliary feature is the cosine similarity between the *tf-idf* vectors of the words in the mention span and the words in the candidate *CUI documents*. For each CUI in UMLS, we created a *pseudo document* which we call the CUI document. The CUI document is a concatenation of the

following information from UMLS:

1. CUI ID and label; for example, *C0000970 (acetaminophen)*
2. Names of the types of the CUI; e.g., *Organic Chemical; Pharmacologic Substance*
3. Definition text for the CUI; e.g., *analgesic antipyretic derivative of acetanilide; weak antiinflammatory properties and is used as a common analgesic, but may cause liver, blood cell, and kidney damage*.
4. All variants for the CUI; e.g., *Acetaminophen, Paracetamol*
5. Select semantic relations between the CUI under consideration and other CUIs; for example, *(may_treat: fever), (may_treat: pain)*.

The intuition behind using this feature is that the span would have a greater lexical overlap with a CUI document that it links to and thus have a higher similarity score.

3.2.3 Context-CUI document similarity

This auxiliary feature is very much like the span-CUI document similarity feature. For this feature as well, we use the *pseudo* CUI documents created using UMLS. However, instead of using the span for calculating the similarity we use the entire context surrounding the span. In the earlier example, the entire sentence "The patient reports suffering from back pain for several years prior to treatment" is the context. We note that for short documents, the context may be the entire document that contains the span to be linked. This means that some unique spans could have the same context. The context-CUI document similarity is the cosine similarity between the *tf-idf* vectors of words in the context and words in the CUI document.

3.2.4 Word embeddings

The auxiliary features discussed so far only capture the superficial lexical aspects of the data used for ensembling. The word embeddings features capture the semantic dimension of the data. We trained the continuous bag of words model (Mikolov et al., 2013) on the entire UMLS knowledge base with word vector dimension of 200 and window-size of 10. Ling et al. (2015) show that these parameters enable capturing long range dependencies. In this way we obtain a vector representation for every word in UMLS. We note that we chose the UMLS corpus as opposed to medical documents so as to have better CUI coverage.

We used these word vectors to create the CUI document vector representation in the following way. Recall that the CUI document is a pseudo document made up of information about the CUI in UMLS. In order to obtain the embedding for a context, span or document, we use the technique described in (Le and Mikolov, 2014). We add up all the embedding vectors representing the words in the CUI document and normalize the sum by the number of words. The resultant vector represents the CUI document embedding. Similarly, we also obtain the span and the context embeddings by adding and normalizing the vectors representing the words in the span and context respectively. Note that if a word in the span or context does not have a vector representation then we just ignore it. Finally, we measure the cosine similarity between the span-CUI document and context-CUI document embedding vectors and use it as a feature for our classifier. Representing the concepts in vector space enables the stacker to learn deep semantic patterns for cases where just lexical information is not sufficient.

4 Experimental Results

4.1 Baselines

We compare our approach to several supervised and unsupervised baselines. The first is *Union* which accepts all predictions for all systems to maximize recall. It classifies all span-CUI links as correct and always includes them.

The second baseline is *Voting*. For this approach, we vary the threshold on the number of systems that must agree on a span-CUI link from one to all. This gradually changes the system behavior from union to intersection of the links. We identify the threshold that results in the highest F1 score on the training dataset. We use this threshold for the voting baseline on the test dataset.

The third baseline is an *oracle threshold* version of Voting. Since the best threshold on the training data may not necessarily be the best threshold for the test data, we identify the best threshold for the test data by plotting a precision-recall curve and finding the best F1 score for the voting baseline. Note that this gives an upper bound on the best results that can be achieved with voting, assuming an optimal threshold is chosen. Since the upper bound can not be predicted without using the test

dataset, this baseline has an unfair advantage.

In addition to the above common baselines, we also compare our approach to a state-of-the-art ensembling system, Bipartite Graph based Consensus Maximization (BGCM) (Gao et al. (2009)). In addition to the output of supervised models, this ensembling technique uses unsupervised models to provide additional constraints and evidence to the classification algorithm. The rationale behind this approach is that objects that are in the same cluster should be more likely to receive the same class label compared to the objects in different clusters. The objective is to predict the class label of an instance in a way that favors agreement between supervised components and at the same time satisfies the constraints enforced by the clustering models. BGCM ensembles multiple models by performing an optimization over a bipartite graph of systems and outputs.

4.2 Dataset Description

All systems and baselines were evaluated on three datasets. Scores reflect the quality of concepts assigned to text spans, as decided by human judges. Detecting span boundaries is not part of this evaluation – all systems are given the same span as input. Annotations were performed by several human judges. For scoring, each text span was paired with a list of concepts produced by all component systems. Annotators marked each span-concept pair correct or incorrect.

The *MCR* dataset (Aggarwal et al., 2015) resulted from running a CRF-based entity recognition system that extracted 1,570 clinical factors from 100 short descriptions (averaging 8 sentences, 100 words) of patient scenarios. The annotated dataset contains a subset of 400 spans resulting in 6,139 annotated span-CUI pairs. The average of the pairwise kappa scores for annotator agreement on the *MCR* dataset was 0.56.

The *i2b2* dataset (Uzuner et al., 2011) is based on the annotated patient discharge summaries released with the 2010 i2b2/VA challenge. The concept extraction task was to identify and extract the text span corresponding to patient medical problems, treatments and tests in unannotated patient record text. We created an entity linking dataset from a random subset of 100 annotated text spans. We ran all available entity linking systems and produced 2,224 annotated span-CUI pairs. The average pairwise kappa score for annotator agreement on the *i2b2* dataset was 0.52.

The Electronic Medical Record dataset (*EMR*) is a private dataset containing spans of medical terms identified in doctors' notes within patient medical records. This dataset has 350 text spans with 3,991 annotated span-CUI pairs. Annotators for the *EMR* dataset reconciled their annotations to build the ground truth.

4.3 Evaluation Metrics

As noted in section 1, *UMLS* often has multiple distinct CUIs for the same semantic concept. So for a given span from a dataset, there may be many true positive concepts in the ground truth. This leads to two possible scoring schemes: *CUI level* and *Span level*. For CUI level scoring, every CUI in the ground truth is a ground truth positive instance. A CUI produced by the Entity Linking system for a given span is a true positive if it is in the ground truth for that span and a false positive if it is not. CUIs in the ground truth for the span that are not produced by the system are counted as false negatives. Spans that have many CUIs in the ground truth, therefore, will have more weight in the precision and recall than spans with fewer CUIs. But since the number of appropriate CUIs for a span is often a side effect of the imperfect automatic merging of concepts in building *UMLS*, the bias is unnatural.

An alternative scoring scheme awards only one true positive, false positive or false negative for each span, not each CUI. For this span level scoring, we report two versions of the metrics. The first version, which we call "Factor Level" in the reported results, aggregates CUI scores using *MAX*. The system scores a true positive if *any* of the CUIs it produces are in the ground truth for the span. It scores a false positive if *none* of its CUIs are in the ground truth. It scores a false negative if it produces no CUIs and there is at least one CUI in the ground truth.

The second version of span level scoring accounts for the fact that the system may produce a mixture of correct and incorrect CUIs for the same span. Each span still has a weight of one in the overall precision and recall, but the system's score for "true positiveness" and "false positiveness" can be a real number between 0 and 1. We call this scoring scheme "Quantum". The quantum true positive score for a span is the number of CUIs produced by the system that are in the

Approach	CUI Level			Factor Level			Quantum		
	P	R	F1	P	R	F1	P	R	F1
GBMCR	0.349	0.242	0.286	0.395	0.437	0.415	0.357	0.268	0.306
NBMCR	0.414	0.179	0.250	0.463	0.511	0.486	0.423	0.163	0.236
VBMCR	0.496	0.215	0.300	0.548	0.605	0.575	0.513	0.198	0.285
CFV	0.587	0.405	**0.479**	**0.903**	0.461	0.611	**0.716**	0.188	0.298
CTakes	0.384	0.245	0.299	0.711	0.577	0.637	0.498	0.202	0.287
MetaMap	0.447	0.219	0.293	0.623	0.652	0.637	0.535	0.215	0.306
CMap	0.179	**0.549**	0.270	0.802	**0.870**	**0.834**	0.305	**0.461**	**0.367**
STR	**0.623**	0.217	0.322	0.623	0.688	0.654	0.623	0.217	0.322
Union	0.207	**0.797**	0.329	**0.888**	**0.981**	**0.932**	0.278	**0.765**	0.408
Majority Voting	**0.746**	0.182	0.293	0.768	0.522	0.622	**0.745**	0.169	0.275
Oracle Voting	0.626	0.290	0.396	0.723	0.707	0.715	0.629	0.251	0.359
BGCM	0.481	0.430	**0.454**	0.753	0.822	0.786	0.525	0.368	**0.433**
Stacking	0.481	0.508	0.494	0.785	0.848	0.815	**0.501**	0.412	0.452
+ CUI Type	0.474	0.573	0.519	0.816	0.889	0.851	0.484	0.502	0.493
+ Span & Context Similarity	0.472	**0.575**	0.519	0.811	0.886	0.847	0.485	**0.508**	0.496
+ CBOW embedding	**0.567**	0.500	**0.532**	**0.824**	**0.892**	**0.857**	0.491	0.507	**0.499**

Table 1: Results on the MCR dataset.

ground truth for the span divided by the total number of CUIs produced by the system (*i.e.*, the span-level Precision). Quantum false positive score is the number of incorrect CUIs produced by the system divided by the total number of CUIs produced.

4.4 Results

We present results for entity linking in the medical domain on the three datasets described in section 4.2 using the evaluation metrics defined in section 4.3. The results include the performance of the individual models, several baselines and various ablations of the auxiliary features using stacking. Tables 1, 2 and 3 show performance on the *MCR*, *i2b2* and *EMR* datasets respectively.

Although we observe similar trends across all the datasets, no single individual model performs better than others across all the evaluation metrics. This led us to conclude that each individual model is optimized for a particular type of entity or data. For example, a model that is good at linking medical drugs might not perform as well on linking medical diseases. In order to leverage the strengths of each individual model, we ensemble them into one powerful model that works across all datasets as well as different evaluation metrics.

As expected, the *Union* baseline obtains the best recall and *Majority Voting* has the highest precision across all datasets. *Oracle Voting* is optimized for F1 and thus obtains an F1 higher than *Majority Voting*. Vanilla stacking beats the best component and baseline systems' F1 scores for CUI level and quantum metrics on all datasets. Adding each auxiliary feature further boosts the performance and we obtain the highest F1 for all datasets using all the features combined. Stacking outperforms the *BGCM* ensembling baseline on all datasets.

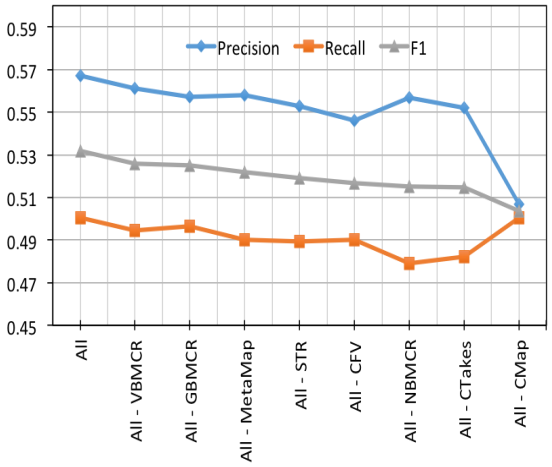

Figure 2: Ablation on the component systems in the ensemble for the *MCR* dataset using the CUI level metric. The systems are arranged in decreasing order of F1 score.

For a deeper understanding of the results, we performed ablation tests on the systems used in the final ensemble. Figure 2 shows the performance of the ensemble with each component ablated in turn. This experiment shows that every component system contributes to the ensemble in either precision, recall or both. While each component contributes to the overall performance, the strength of the ensemble is determined by the combination of

Approach	CUI Level			Factor Level			Quantum		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
GBMCR	0.507	0.375	0.431	0.790	0.807	0.798	0.515	0.427	0.467
NBMCR	0.478	0.356	0.408	0.779	0.796	0.787	0.486	0.403	0.441
VBMCR	0.554	0.404	**0.467**	0.800	0.817	0.809	0.564	**0.468**	**0.511**
CFV	0.173	**0.457**	0.251	0.884	0.903	0.894	0.577	0.327	0.417
CTakes	0.564	0.213	0.309	0.861	0.731	0.791	**0.677**	0.195	0.303
MetaMap	0.565	0.154	0.242	0.750	0.742	0.746	0.647	0.153	0.248
CMap	0.216	0.360	0.270	**0.894**	**0.903**	**0.898**	0.410	0.260	0.318
STR	**0.825**	0.176	0.290	0.833	0.860	0.847	0.566	0.236	0.333
Union	0.191	**0.855**	0.312	**0.969**	**1.000**	**0.984**	0.352	**0.849**	**0.498**
Majority Voting	**0.705**	0.189	0.298	0.846	0.828	0.837	**0.766**	0.176	0.286
Oracle Voting	0.624	0.270	0.373	0.874	0.893	0.883	0.709	0.227	0.344
BGCM	0.469	0.406	**0.435**	0.938	0.968	0.952	0.509	0.386	0.439
Stacking	0.434	0.697	0.535	**0.958**	**0.989**	**0.974**	0.481	0.655	0.555
+ CUI Type	0.525	0.730	0.611	0.927	0.957	0.942	**0.547**	0.563	0.555
+ Span & Context Similarity	0.528	0.756	0.622	0.927	0.957	0.942	0.544	0.639	0.588
+ CBOW embedding	**0.528**	**0.756**	**0.622**	0.938	0.968	0.952	0.546	**0.700**	**0.607**

Table 2: Results on the *i2b2* dataset.

Approach	CUI Level			Factor Level			Quantum		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
GBMCR	0.338	0.134	0.192	0.369	0.351	0.36	0.360	0.315	0.196
NBMCR	0.381	0.151	0.217	0.410	0.390	0.400	0.396	0.148	0.216
VBMCR	0.564	0.224	0.321	0.618	0.589	0.603	0.600	0.225	0.327
CFV	0.510	0.353	**0.417**	**0.914**	0.607	0.729	0.692	0.249	0.366
CTakes	0.403	0.321	0.357	0.706	0.628	0.665	0.527	0.268	0.355
MetaMap	0.460	0.220	0.298	0.575	0.568	0.571	0.527	0.223	0.313
CMap	0.205	**0.597**	0.305	0.761	**0.766**	**0.763**	0.334	**0.597**	**0.428**
STR	**0.714**	0.284	0.406	**0.714**	0.748	0.730	0.714	0.284	0.406
Union	0.187	**0.739**	0.299	0.857	**0.852**	**0.854**	0.272	**0.676**	0.388
Majority Voting	**0.879**	0.225	0.359	**0.912**	0.561	0.695	**0.894**	0.220	0.353
Oracle Voting	0.668	0.297	0.412	0.820	0.661	0.732	0.719	0.276	0.399
BGCM	0.453	0.419	**0.435**	0.801	0.809	0.805	0.482	0.409	**0.442**
Stacking	0.443	0.517	0.477	0.794	**0.832**	**0.812**	0.488	**0.463**	0.475
+ CUI Type	0.559	0.548	0.554	0.807	0.778	0.792	0.571	0.436	0.495
+ Span & Context Similarity	0.593	**0.554**	0.573	0.820	0.781	0.800	0.616	0.443	0.515
+ CBOW embedding	**0.667**	0.549	**0.602**	**0.830**	0.775	0.801	**0.669**	0.439	**0.530**

Table 3: Results on the *EMR* dataset.

the component systems. The ablation of the CMap system has the highest impact on the ensemble, reducing the F1 score by 5.2%. We obtained similar plots for the factor level and quantum metrics and we expect to see similar trends for the *i2b2* and the *EMR* datasets as well.

5 Discussion

The experimental results presented in section 4.4 confirm that the different component systems show significantly different behavior on different metrics for different datasets. No individual system was universally the best. *CMap* had consistently good Recall but low Precision. *CFV* scored well in certain circumstances on precision, recall and F1 score, but this varied from dataset to dataset and metric to metric. *STR* usually had relatively high precision, but low recall, and *VBMCR* had very good F1 scores on *i2b2*, but was less impressive on the other datasets.

These observations imply good conditions for ensembling to make a difference. Even so, the best baseline ensemble only outperforms the best component system on F1 in four of the nine experiments (metric-dataset combinations). Stacking outperforms the best component system in all nine, and outperforms the best ensembling baseline for six of the nine – all of the CUI level metrics and quantum, but never at the factor level. The factor level scoring is much more generous, but it is not immediately clear why this would benefit naïve ensembling over stacking.

Auxiliary features almost always improve stacking. Again the exception is with factor level scoring. Interestingly, auxiliary features almost universally improve precision significantly without too damaging an effect on recall. This result suggests that it would be worthwhile experimenting with the precision-vs-recall bias of component systems to see if Stacking with auxiliary features could be used, for example, to recover precision with recall-biased components.

6 Related Work

The problem of entity linking has received considerable attention in the research community. Several community tasks are focused specifically on the medical domain and are addressing the problem of linking disease/disorder entities to *SNOMED CT*.[4] *SNOMED CT* concepts are also included in *UMLS*.

The ShARe/CLEF eHealth Evaluation Lab 2013 (Suominen et al., 2013) consists of a collection of tasks focused on facilitating patients' understanding of their medical discharge summaries. The assumption is that an improved understanding of medical concepts in such documents can be achieved by normalizing all health conditions to standardized *SNOMED CT* concepts. Using these concepts, the medical documents can further be connected to other patient friendly sources.

The Open Biomedical Annotator (*OBA*) (Jonquet et al., 2009) is an ontology-based Web service that annotates public datasets with biomedical ontology concepts, including concepts from *UMLS*. The *OBA* is based on dictionary matching. The dictionary is a list of strings that identify ontology concepts. The dictionary is constructed by accessing biomedical ontologies and extracting all concept names, their synonyms or labels. The web service takes as input the user's free text. The tool recognizes concepts using string matching on the dictionary and outputs the concept annotations.

There are several notable approaches to perform entity linking in the open domain. These open domain approaches often deal with named entities. The linking targets in this case are often single, unambiguous, specific concepts. The problem of finding domain-specific concepts, on the other hand, can be more challenging as there may be appropriate concepts at different levels of specificity, and concepts are more compositional and contextual. Approaches such as DBPedia Spotlight (Mendes et al., 2011) and AIDA (Hoffart et al., 2011) use Wikipedia to find the links of recognized entity mentions.

To overcome challenges of obtaining labeled medical datasets, Zheng et al. (2015) proposed an unsupervised approach for entity linking. More traditional sieve-based techniques have been used for this task recently (D'Souza and Ng, 2015).

Using ensembling techniques for open domain entity linking has shown good performance in the past (Rajani and Mooney, 2017) on the Trilingual Entity Discovery and Linking (TEDL) task. TEDL is an entity linking task conducted by NIST. The goal of this task is to discover entities in the three included languages (English, Spanish and Chinese) from a supplied text corpus and link these entities to an existing English knowledge base (a reduced version of FreeBase).

Rajani and Mooney (2016) proposed an approach for combining multiple supervised and unsupervised models for entity linking. Their technique improves the previous result on the TEDL task. Another ensembling approach is Mixtures of Experts (Jacobs et al., 1991) which employs divide-and-conquer principle to soft switch between learners covering different sub-spaces of the input using Expectation-Maximization (EM). Our work is the first we know of to use ensembling for entity linking in the medical domain.

7 Conclusion

We have identified an entity linking task in the medical domain for which existing technologies perform differently on different metrics for different datasets. Such an environment presents an obvious opportunity for ensembling techniques.

We have built a stacking ensembler using multiple diverse entity linking systems. The auxiliary features further boost the stacker's performance. Experiments confirm that naïve ensembling does not always outperform component entity linking systems, but that vanilla stacking does. Adding auxiliary features to the stacker almost universally improves its precision without harming recall, giving it generally the best F1 scores overall.

Our model is able to fuse additional relevant knowledge from multiple systems and leverage them to improve prediction.

[4] SNOMED CT: http://www.snomed.org/

References

Nitish Aggarwal, Ken Barker, and Chris Welty. 2015. Medical concept resolution. In *Proceedings of the ISWC 2015 Posters & Demonstrations Track co-located with the 14th International Semantic Web Conference (ISWC-2015), Bethlehem, PA, USA, October 11, 2015.*.

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*. Springer-Verlag, pages 1–15.

Jennifer D'Souza and Vincent Ng. 2015. Sieve-based entity linking for the biomedical domain. In *ACL (2)*. pages 297–302.

Jing Gao, Feng Liang, Wei Fan, Yizhou Sun, and Jiawei Han. 2009. Graph-based consensus maximization among multiple supervised and unsupervised models. In *NIPS2009*. pages 585–593.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the Seventh International Conference on Semantic Systems*. pages 1–8.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation* 3(1):79–87.

Clement Jonquet, Nigam Shah, and Mark Musen. 2009. The open biomedical annotator. pages 56–60.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the International Conference on Machine Learning*. volume 14, pages 1188–1196.

Wang Ling, Lin Chu-Cheng, Yulia Tsvetkov, and Silvio Amir. 2015. Not all contexts are created equal: Better word representations with variable attention. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP-15)*. Citeseer.

Pablo N. Mendes, Max Jakob, Andrés Garcia-Silva, and Christian Bizer. 2011. DBpedia spotlight: shedding light on the web of documents. In *Proceedings of the Seventh International Conference on Semantic Systems*. pages 1–8.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Nazneen Fatema Rajani and Raymond J. Mooney. 2016. Combining Supervised and Unsupervised Ensembles for Knowledge Base Population. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP-16)*.

Nazneen Fatema Rajani and Raymond J. Mooney. 2017. Stacking With Auxiliary Features. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI2017)*. Melbourne, Australia.

Hanna Suominen, Sanna Salanterä, Sumithra Velupillai, Wendy W. Chapman, Guergana Savova, Noemie Elhadad, Sameer Pradhan, Brett R. South, Danielle L. Mowery, Gareth J. F. Jones, Johannes Leveling, Liadh Kelly, Lorraine Goeuriot, David Martinez, and Guido Zuccon. 2013. *Overview of the ShARe/CLEF eHealth Evaluation Lab 2013*, pages 212–231.

Ozlem Uzuner, Brett R. South, Shuying Shen, and Scott L. DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *JAMIA* 18(5):552–556.

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

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

Jin G Zheng, Daniel Howsmon, Boliang Zhang, Juergen Hahn, Deborah McGuinness, James Hendler, and Heng Ji. 2015. Entity linking for biomedical literature. *BMC medical informatics and decision making* 15(1):S4.