Stacked Generalization for Medical Concept Extraction from Clinical Notes

Youngjun Kim
School of Computing
University of Utah
Salt Lake City, UT 84112
youngjun@cs.utah.edu

Ellen Riloff
School of Computing
University of Utah
Salt Lake City, UT 84112
riloff@cs.utah.edu

Abstract

The goal of our research is to extract medical concepts from clinical notes containing patient information. Our research explores stacked generalization as a meta-learning technique to exploit a diverse set of concept extraction models. First, we create multiple models for concept extraction using a variety of information extraction techniques, including knowledge-based, rule-based, and machine learning models. Next, we train a meta-classifier using stacked generalization with a feature set generated from the outputs of the individual classifiers. The meta-classifier learns to predict concepts based on information about the predictions of the component classifiers. Our results show that the stacked generalization learner performs better than the individual models and achieves state-of-the-art performance on the 2010 i2b2 data set.

1 Introduction

Clinical notes (or electronic medical records) contain important medical information related to patient care management. Health care professionals enter a patient's medical history and information about their care at a health care provider. A patient's diseases, symptoms, treatments, and test results are often encoded in these notes in an unstructured manner.

In the last two decades, Natural Language Processing (NLP) techniques have been applied to clinical notes for medical concept extraction. Medical concept extraction typically consists of two main steps: detection of the phrases that refer to medical entities, and classification of the semantic category for each detected medical entity. Medical domain knowledge and sophisticated information extraction methods are required to achieve high levels of performance. Medical concept extraction is a fundamental problem that can also serve as the stepping stone for higher level tasks, such as recognizing different types of relationships between pairs of medical concepts.

The main goal of our research is to explore the use of stacked generalization learning for the medical concept extraction task. Stacked learning (Wolpert, 1992) is a meta-learning ensemble-based method that regulates the biases of multiple learners and integrates their diversities. An ensemble of individual classifiers is created and then another classifier (the meta-classifier) sits on top of the ensemble and trains on the predictions of the component classifiers. A key advantage of stacked generalization is that the meta-classifier learns how to weight and combine the predictions of the individual classifiers, allowing for a fully automated ensemble system. New component classifiers can be easily added without the need for manual intervention. Voting-based ensembles are another strategy for combing multiple classification models, and they often perform well. But they can require manual adjustment of the voting threshold when new components are added, and they do not automatically learn how to weight different components. Stacked generalization provides a more easily extensible and adaptable framework.

In the next sections, we discuss related work, describe our individual classifiers for medical concept extraction, and present the stacked generalization learning framework. Finally, we present experimental results on the 2010 i2b2 data set and compare our results with state-of-the-art systems.

2 Related Work

In early natural language processing (NLP) research for clinical notes, most systems used rule-based approaches. MedLEE (Friedman et al., 1994) uses a rule-based system that extracts med-

Proceedings of the 2015 Workshop on Biomedical Natural Language Processing (BioNLP 2015), pages 61–70,
Beijing, China, July 30, 2015.

ical concepts by performing a shallow syntactic analysis and using semantic lexicons. SymText was developed by Haug et al. (1995; 1997) and evolved into MPlus (Christensen et al., 2002). This system was used to extract medical findings, diseases, and appliances from chest radiograph reports. HITEx (Zeng et al., 2006) is a pipelined system with multiple preprocessing modules and has been used to extract family history information, principal diagnosis, comorbidity and smoking status from clinical notes. MetaMap (Aronson and Lang, 2010) was developed to recognize Metathesaurus concepts from biomedical texts by utilizing the UMLS (Unified Medical Language System).

Recently, statistical learning approaches have received more attention because of the manual effort typically required to create rule-based systems. Most current information extraction (IE) systems in clinical NLP use statistical machine learning approaches that often achieve better performance than rule-based approaches. Our work is also closely related to Named Entity Recognition (NER). For both newswire and biomedical texts, machine learning models have achieved good results for extracting specific types of entities (e.g., (Collier et al., 2000; Lafferty et al., 2001; Collins, 2002; Zhou and Su, 2002; McDonald and Pereira, 2005)).

Our research focuses on the medical concept detection task that was introduced in 2010 for the *i2b2 Challenge Shared Tasks* (Uzuner et al., 2011). These challenge tasks included: (a) the extraction of medical problems, tests, and treatments, (b) classification of assertions made on medical problems, and (c) relations between medical problems, tests, and treatments. The best performance on the 2010 i2b2 concept extraction task (a) was achieved by de Bruijn et al. (2011) with 83.6% recall, 86.9% precision, and 85.2% F_1 score. They integrated many features commonly used in NER tasks including syntactic, orthographic, lexical, and semantic information (from various medical knowledge databases). Jiang et al. (2011) trained a sequence-tagging model that consisted of three components in a pipeline: concept taggers with local features and outputs from different knowledge databases, post-processing programs to determine the correct type of semantically ambiguous concepts, and a voting ensemble module to combine the results of different taggers. Their system achieved an 83.9% F_1 score. Subsequent research by Tang et al. (2013) showed that clustering and distributional word representation features achieved an higher F_1 score of 85.8%.

Ensemble methods that combine multiple classifiers have been widely used for many NLP tasks and generally yield better performance than individual classifiers. For protein/gene recognition, Zhou et al. (2005) used majority voting from multiple classifiers to achieve better performance than any single classifier. Finkel et al. (2005) combined the outputs of forward and backward (reversing the order of the words in a sentence) sequence labelling, which improved recall. Similarly, Huang et al. (2007) integrated the outputs of three models for gene mention recognition. They applied intersection to the outputs of forward and backward labeling SVM (support vector machine) models and then union with the outputs of one CRF (conditional random fields) model. Doan et. al (2012) showed that a voting ensemble of rule-based and machine learning systems obtained better performance than individual classifiers for medication detection. For medical concept detection, Kang et al. (2012) used majority voting between seven different systems for performance improvement.

Our research explores an ensemble method called stacked generalization (Wolpert, 1992; Breiman, 1996), which has been shown to produce good results for several NLP tasks. Stacking is an ensemble-based method for combining multiple classifiers by training a meta-classifier using the outputs of the individual classifiers. Ting and Witten (1999) showed that stacked generalization using confidence scores from the predictions of multiple classifiers obtained better results than the individual systems. Džeroski and Zeno (2004) showed good performance for stacked learning on a collection of 21 datasets from the UCI Repository of machine learning databases (Blake and Merz, 1998). Nivre and McDonald (2008) applied stacked learning to dependency parsing by integrating two different models (graph-based models and transition-based models). Recently, some research has used stacked learning in the bioinformatics domain. Wang et al. (2006) used stacked learning with two base learners for predicting membrane protein types. Netzer et al. (2009) applied stacked generalization to identify breath gas marker and reported improved classification accuracy. For NLP from clinical texts, Kilicoglu et al. (2009) used stacked learning for document level

classification to identify rigorous, clinically relevant studies.

Stacked learning is similar to weighted majority voting (Littlestone and Warmuth, 1994) and Cascading learning (Gama and Brazdil, 2000). However, weighted majority voting only determines a voting weight for each individual classifier, while stacked learning can assign different weights to different types of predictions. Training in cascading learning requires multiple rounds of learning, while stacked learning typically consists of just two stages. Also, cascading learning does not need multiple base learners. Tsukamoto et al. (2002) employed cascaded learning using a single algorithm that improved performance on an NER task.

Our stacked generalization framework is different from weighted majority voting or cascading learning. Our stacked learning architecture trains a meta-classifier using features derived from the predictions and confidence scores of a set of diverse component classifiers. To the best of our knowledge, this research is the first to use stacked generalization with a rich set of meta-features for medical concept extraction from clinical notes.

3 Stacked Generalization with Multiple Concept Extraction Models

The goal of our research is to investigate stacked generalization learning for medical concept extraction with a diverse set of information extraction models. We will first describe each individual model and then present the stacked learning framework.

3.1 Information Extraction Models

Our ensemble consists of four types of individual component systems, which are described below.

MetaMap: We use a widely-used knowledge-based system called MetaMap (Aronson and Lang, 2010). MetaMap is a rule-based program that assigns UMLS Metathesaurus semantic concepts to phrases in natural language text. Unlike our other IE systems, MetaMap is not trained with machine learning so it is not dependent on training data. Instead, MetaMap is a complementary resource that contains a tremendous amount of external medical knowledge.

We encountered one issue with using this resource for our task. MetaMap can assign a large set of semantic categories, many of which are not relevant to the i2b2 concept extraction task. However it is not obvious how to optimally align the MetaMap semantic categories with our task's semantic categories because their coverage can substantially differ. Therefore we built a statistical model based on the concepts that MetaMap detected in the training data. We collected all of MetaMap's findings in the training data, aligned them with the gold standard medical concepts, and calculated the probability of each MetaMap semantic category mapping to each of our task's three concept types ("problem", "treatment", and "test"). We then assigned a MetaMap semantic type to one of our concept types if the semantic type is ranked among the top 30% of semantic types based on Prob(concept_type | sem_type). For example, "sosy" ("Sign or Symptom" in MetaMap) was mapped to the "problem" concept type because it had a high probability of being aligned with labeled problems in the data set. Table 1 shows the semantic types that we ultimately used for concept extraction.[1]

Category	MetaMap semantic types
Problem	acab, anab, bact, celf, cgab, chvf, dsyn, inpo, mobd, neop, nnon, orgm, patf, sosy
Treatment	antb, carb, horm, medd, nsba, opco, orch, phsu, sbst, strd, topp, vita
Test	biof, bird, cell, chvs, diap, enzy, euka, lbpr, lbtr, mbrt, moft, phsf, tisu

Table 1: MetaMap semantic types used for concept extraction.

Rules: We used the training data to automatically create simple rules. The idea is to exploit the training data to create a simple rule-based system without any manual effort. For each phrase labeled as a medical concept in the training data, we created a rule that maps the phrase to the concept type that it was most frequently assigned to in the training data. Similar to the MetaMap model above, we then computed P(concept_type | phrase) using frequency counts.

To generate phrase matching rules, we applied

[1]Refer to `http://metamap.nlm.nih.gov/Docs/SemanticTypes_2013AA.txt` for the mapping between abbreviations and the full semantic type names.

two thresholds to each rule: a minimum probability threshold (θ_P) and a minimum frequency threshold (θ_F). First, we extracted annotated phrases from the training data. Next, for each phrase we computed its overall frequency and P(concept_type | phrase) for each of the 3 concept types. We then selected the phrases that passed the two thresholds and assigned them to the corresponding concept type. In cases where one phrase subsumed another phrase, such as "disease" and "coronary disease", and both phrases pass the thresholds, we only chose the longer phrase. We then created a rule for each phrase that labels all instances of that phrase as the concept type (e.g., "diabetes" $\rightarrow Problem$). A concept was extracted when the candidate phrase occurs more than two times (θ_F) in the training data and the rule's probability is over 60% (θ_P).

Contextual Classifier (SVM): We created a supervised learning classifier with contextual features. We applied the Stanford CoreNLP tool (Manning et al., 2014) to our data sets for tokenization, lemmatization, part-of-speech (POS) tagging, and Named Entity Recognition (NER). We trained a Support Vector Machine (SVM) classifier with a linear kernel using the LIBLINEAR (Library for Large Linear Classification) software package (Fan et al., 2008) for multi-class classification.

We reformatted the training data with IOB tags (B: at the beginning, I: inside, or O: outside of a concept). We defined features for the targeted word's lexical string, lemma, POS tag, affix(es), orthographic features (e.g. Alphanumeric, HasDigit), named entity tag, and pairwise combinations of these features. Also, we used the predictions of MetaMap as additional features. Table 2 shows the complete feature set used to create the SVM model, as well as the CRF models described below. We set the cost parameter to be $c = 0.1$ (one of LIBLINEAR's parameters) after experimenting with different values by performing 10-fold cross validation on the training set.

Sequential Classifier (CRF): We trained several sequential taggers using linear chain Conditional Random Fields (CRF) supervised learning models. In contrast to the contextual classifier mentioned above, the CRF classifiers use a structured learning algorithm that explicitly models transition probabilities from one word to the next. Our CRF models also use the features in Table 2. We used Wapiti (Lavergne et al., 2010), which is a simple and fast discriminative sequence labeling toolkit, to train the sequential models. As with the SVM, 10-fold cross validation was performed on the training set to tune the Wapiti's CRF algorithm parameters. We set the size of the interval for the stopping criterion to be $e = 0.001$. For regularization, $L1$ and $L2$ penalties were set to 0.005 and 0.4 respectively.

Feature	Description
Word	w_0 (current word), w_{-1} (previous word), w_1 (following word), w_{-2} (second previous word), w_2 (second following word)
Bi-grams of words	$[w_{-2}, w_{-1}]$, $[w_{-1}, w_0]$, $[w_0, w_1]$, $[w_1, w_2]$
Lemmas	$l_{-3}, l_{-2}, l_{-1}, l_1, l_2, l_3$
Affixes	prefixes and suffixes, up to a length of 5
Orthographic	15 features based on regular expressions for w_0, w_{-1}, w_1
POS tags	$p_0, p_{-1}, p_1, p_{-2}, p_2$
Bi-grams of POS tags	$[p_{-2}, p_{-1}]$, $[p_{-1}, p_0]$, $[p_0, p_1]$, $[p_1, p_2]$
Lemma + POS	$[l_0, p_0]$
NER class	n_0
MetaMap semtype	m_0, m_{-1}, m_1, $[m_{-1}, m_0]$, $[m_0, m_1]$

Table 2: Feature set for SVM and CRF models.

Post processing: The concepts annotated by the i2b2 annotation guidelines[2] include modifying articles, pronouns, and prepositional phrases. For treatments such as medications, the amount, dose, frequency, and mode are included in the annotation only when they occur as pre-modifiers. However, when they are part of *signatura*, which explains how to use the medication for the patient, they are excluded from concept boundaries. For example,

800 mg ibuprofen
Lasix 20 mg b.i.d. by mouth

[2] https://www.i2b2.org/NLP/Relations/assets/ConceptAnnotationGuideline.pdf

"800 mg ibuprofen" is annotated as a treatment concept, while only "Lasix" is annotated in the second example.

When applying MetaMap to the training set, we observed that there is a huge difference between the i2b2 annotations and MetaMap's concept boundary definition, especially with respect to articles and pronouns. MetaMap typically excludes modifying articles, pronouns, and prepositional phrases. For example, for "a cyst in her kidney", only "cyst" was extracted by MetaMap.

Therefore we added a post-processing step that uses three simple heuristics to adjust concept boundaries to reduce mismatch errors. Although these rules were originally compiled for use with MetaMap, we ultimately decided to apply them to all of the IE models. The three heuristic rules are:

1. We include the preceding word contiguous to a detected phrase when the word is a quantifier (e.g., "some"), pronoun (e.g., "her"), article (e.g., "the'), or quantitative value (e.g., "70%").
2. We include a following word contiguous to a detected phrase when the word is a closed parenthesis (")") and the detected phrase contains an open parenthesis ("(").
3. We exclude the last word of a detected phrase when the word is a punctuation mark (e.g., period, comma).

3.2 Ensemble Methods

We explored two types of ensemble architectures that use the medical concept extraction methods described above as components of the ensemble. We created a Voting Ensemble, as a simple but often effective ensemble method, and a Stacked Generalization Ensemble, which trains a meta-classifier with features derived from the outputs of its component models. Both architectures are described below.

Voting Ensemble Method: We implemented the majority voting strategy suggested by Kang et al. (2012) with a simple modification to avoid labeling concepts with overlapping text spans. When two different concepts have overlapping text spans, the concept that receives more votes is selected. For overlapping concepts with identical vote counts, we used the normalized confidence scores from the individual classifiers and select the concept with the higher confidence score. Each confidence score, $s \in S$ (the set of all confidence scores), was normalized by Z-score as:

$$Nor(s) = \frac{s - E(S)}{std(S)} \text{ where}$$

$$E(S) = \text{the mean of the scores}$$

$$std(S) = \text{the standard deviation of the scores}$$

Stacked Generalization Method: We created a meta-classifier by training a SVM classifier with a linear kernel based on the predictions from the individual classifiers. Figure 1 shows the architecture of our stacked learning ensemble. First, to create training instances for a document, all of the concept predictions from the individual IE models are collected. We then use a variety of features to consider the degree of agreement and consistency between the IE models. Each concept predicted by an IE model is compared with all other concepts predicted in the same sentence. For each pair of concepts, the following eight matching criteria are applied to create eight features:

- If the text spans match
- If the text spans partially match (any word overlap)
- If the text spans match and concept types match
- If the text spans partially match and the concept types match
- If the text spans have the same start position
- If the text spans have same end position
- If one text span subsumes the other
- If one text spans is subsumed by the other

Features are also defined that count how many different models produced a predicted concept, and features are defined for predictions produced by just a single model (indicating which model produced the predicted concept).

In addition, we created a feature for the confidence score of each predicted concept. When multiple components predicted a concept, the highest score was used. We also created a feature that counts how many times the same phrase was predicted to be a concept in other sentences in the same document. The number of word tokens in a prediction, and whether the prediction contains a conjunction or prepositional phrase, were also used as features.

We performed 10-fold cross validation on the training set to obtain predictions for each classifier. These predictions were used to train the meta-classifier.

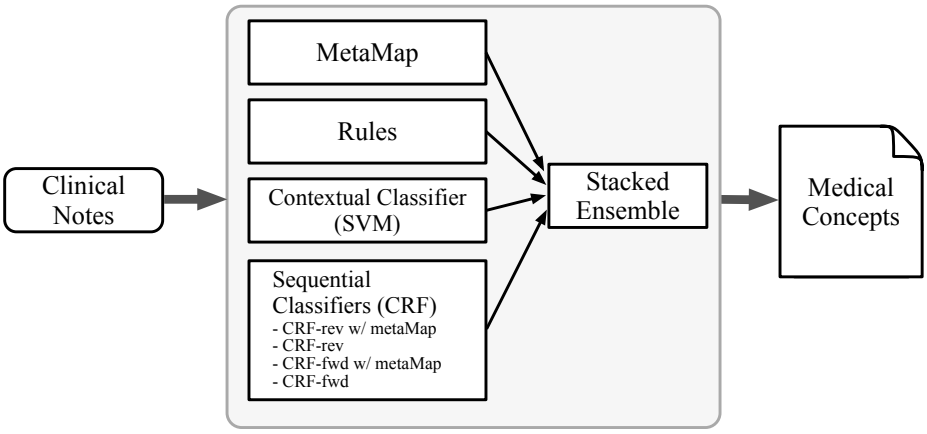

Figure 1: Stacked Learning Ensemble Architecture

4 Experimental Results

We present experimental results for each of our concept extraction components individually, as well as for each of the two ensemble methods: voting and stacked generalization learning.

4.1 Data

The 2010 i2b2 Challenge corpus was used for evaluation. The corpus consists of discharge summaries from Partners HealthCare (Boston, MA) and Beth Israel Deaconess Medical Center, as well as discharge summaries and progress notes from the University of Pittsburgh Medical Center (Pittsburgh, PA). It contains 349 clinical notes as training data and 477 clinical notes as test data. 18,550 problems, 13,560 treatments and 12,899 tests (for a total of 45,009 medical concepts) are annotated as the semantic concepts in the test data.

4.2 Performance of Individual Models

We used the i2b2 Challenge evaluation script to compute recall, precision, and F_1 scores. In this paper, we present the results of class exact match: both the text span and semantic category must exactly match the reference annotation.

MetaMap: We used MetaMap 2013v2 with the 2013AB NLM relaxed database.[3] As we mentioned in Section 3.1, we only used a subset of MetaMap's semantic types based on statistics collected by aligning MetaMap's findings with the medical concepts in the labeled training data.[4] We selected the top 30% of its semantic types (shown in Table 1) based the collected probabilities. The first row of Table 3 shows the results for MetaMap using these semantic categories. As explained before, MetaMap suffers from boundary mismatch errors due to the difference between the i2b2 annotations and MetaMap's concept boundary definition. In spite of our added post-processing rules to address this issue, we could not eliminate this problem especially for concepts containing many pre-modifiers or prepositional phrases. We also observed that MetaMap often did not recognize acronyms and abbreviations in the clinical notes.

Method	Rec	Pr	F
MetaMap	36.1	47.4	41.0
Rules	18.5	72.6	29.5
SVM	81.2	77.5	79.3
CRF-fwd	81.5	86.2	83.8
CRF-fwd w/ MetaMap	82.5	86.7	84.5
CRF-rev	82.4	86.5	84.4
CRF-rev w/ MetaMap	82.9	87.0	84.9
Voting ensemble	83.5	88.2	85.8
Stacked ensemble	**83.5**	**88.6**	**86.0**

Table 3: Recall (Rec), Precision (Pr), and F_1 score (F) of each method on the 2010 i2b2 Challenge test data.

Rules: The second row of Table 3 shows the results of matching with the rules that we extracted from the training data. This simple approach obtained fairly good precision of 72.6%, but low

[3] We used the following MetaMap options: -C -V NLM -y -i -g --composite_phrases 3 --sldi

[4] Using all of MetaMap's semantic types produces extremely low precision.

recall. This method relies entirely on common words found in the training data, so unseen words in the test data were not recognized. In addition, pre-modifiers were often missed. For example, only "embolization" was extracted from text mentioning "coil embolization".

SVM: The SVM context-based classifier achieved an F_1 score of 79.3% (third row in Table 3) with its rich contextual features. A subsequent analysis revealed that this classifier excels at recognizing concepts that consist of a single word, achieving recall of 89.3% for these cases, about 2.3% higher than the sequential classifiers (CRFs) perform on these cases.

CRF: We implemented four different variations of sequential classifiers. We trained CRF classifiers with both forward and backward tagging (by reversing the sequences of words) (Kudo and Matsumoto, 2001; Finkel et al., 2005). As a result, each medical concept had different IOB representations. For example, the IOB tags of "positive lymph nodes" by forward and backward tagging were *"positive/B-problem lymph/I-problem nodes/I-problem"* and *"positive/I-problem lymph/I-problem nodes/B-problem"*, respectively. For each of these forward (CRF-fwd) and backward (CRF-rev) taggers, we created versions both with and without MetaMap output as features. Overall, the CRF models performed better than the other IE methods. Among the four sequential models, backward tagging with MetaMap features obtained the best results, which are shown in row 7 of Table 3, with an F_1 score of 84.9%. A subsequent analysis revealed that this classifier excels at recognizing multi-word concepts, achieving a recall of 79.8% (about 5% higher than the SVM) and a precision of 82.8% (about 7.4% higher than the SVM) for medical concepts with multiple words.

4.3 Performance of Ensembles

Finally, we evaluated the performance of the two ensemble architectures described in Section 3.2.

Voting Ensemble: We created a Voting ensemble consisting of all seven individual IE models: the rules, MetaMap, the contextual classifier, and all four sequential tagging models. The 8th row in Table 3 shows the results with a voting threshold of three (i.e. three votes are needed to label a concept). This voting ensemble obtained better performance than any of the individual classifiers, reaching an F_1 score of 85.8%.

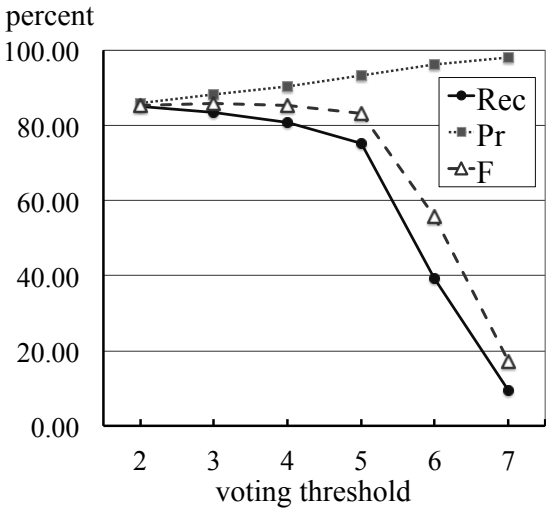

Figure 2: Recall (Rec), Precision (Pr), and F_1 score (F) of the voting ensemble for varying voting thresholds.

The voting threshold is a key parameter for Voting Ensembles that can dramatically affect performance. The voting threshold can serve as a recall/precision knob to obtain different trade-offs between recall and precision. In Figure 2, we show results for voting thresholds ranging from two to seven. The curves show that precision increases as the threshold gets higher, but recall drops simultaneously. When the voting threshold exceeds five, recall drops precipitously.

Stacked Generalization: We evaluated the Stacked Generalization Ensemble using the same set of seven individual IE models used in the Voting Ensemble. The last row of Table 3 shows that the Stacked Ensemble achieved slightly higher precision than the Voting Ensemble, overall producing 83.5% recall, 88.6% precision, and an 86.0% F_1 score. Using a paired t-test across the F_1 scores for all test documents (i.e., each F_1 score was calculated for each document, and then averaged across all test documents), the Stacked Ensemble performed significantly better than all of the individual IE models ($p < 10^{-4}$), but not significantly better than the Voting Ensemble ($p = 0.0849$).

We performed ablation tests for both the Voting and Stacked Generalization Ensembles to evaluate the impact of each IE model on the ensembles. An ablated ensemble was tested by removing a single model from the ensemble. Table 4 shows the F_1 score for each ablated ensemble and the differ-

Method	Voting		Stacked	
	F_1 score	Impact	F_1 score	Impact
MetaMap	85.69	-0.10	85.81	-0.20
Rules	85.76	-0.02	85.93	-0.08
SVM	85.51	-0.28	85.70	-0.31
CRF-fwd	85.56	-0.23	85.84	-0.17
CRF-fwd w/ MetaMap	85.56	-0.22	85.83	-0.18
CRF-rev	85.41	-0.37	85.76	-0.25
CRF-rev w/ MetaMap	85.41	-0.37	85.77	-0.24

Table 4: The ablation tests of Voting and Stacked Generalization Ensembles

ence from the F_1 score of the original (complete) ensemble. As shown in Table 4, every IE model contributed to the performance of both the Voting and Stacked Ensembles. Removing the Rules component had a very small impact, presumably because the machine learning models also acquire information from the training data. All of the other IE models appear to have played a valuable role. For the voting ensemble, the F_1 score dropped the most when the CRF-rev or CRF-rev w/ MetaMap models were removed. For Stacked Generalization, removing the SVM model had the biggest impact.

Overall, our results confirm that ensemble architectures consistently outperform individual IE models. Although the Stacked Ensemble and Voting Ensemble produce similar levels of performance, Stacked Generalization has a significant practical advantage over Voting Ensembles. Adding new models to an ensemble is easy, but Voting Ensembles require a voting threshold that must be adjusted when the number of component models changes. Consequently, it can be difficult to assess the overall impact of adding new models (e.g., adding twice as many models may require a higher voting threshold, which may yield higher precision but substantially lower recall). A simple count-based voting threshold is coarse, so small changes can sometimes produce dramatic effects. In contrast, Stacked Generalization uses a meta-classifier to automatically learn how to best weight and use the components in its ensemble. Consequently, adding new models to a Stacked Ensemble only requires re-training of the meta-classifier.

To demonstrate this advantage over voting, we added a second copy of the MetaMap component as an additional system in our ensemble. Voting between the eight systems using our original threshold of three dropped the F_1 score by -0.3%. Adding a third copy of the MetaMap component (producing nine component systems) decreased the F_1 score by -6.8% (absolute). In the same scenarios, the Stacked Learning Ensemble proved to be much more robust, showing almost no change in performance (-0.2% and -0.3% with eight and nine systems respectively).

Table 5 shows the performance of other state-of-the-art systems for medical concept extraction alongside the results from our Stacked Learning Ensemble. The Stacked Ensemble produces higher precision than all of the other systems. Overall, the F_1 score of the Stacked Ensemble is comparable to the F_1 score of the best previous system by Tang et al. (2013). Our Stacked Ensemble achieves slightly higher precision, while the the Tang et al. system produces slightly higher recall.

System	Rec	Pr	F
de Bruijn et al. (2011)	83.6	86.9	85.2
Kang et al. (2012)	81.2	83.3	82.2
Tang et al. (2013)	84.3	87.4	85.8
Stacked Ensemble	83.5	88.6	86.0

Table 5: Recall (Rec), Precision (Pr), and F_1 score (F) of other state-of-the-art systems and our Stacked Ensemble.

5 Analysis

We did manual error analysis to better understand the nature of the mistakes made by our system. Many of the errors revolved around incorrect boundaries for extracted concepts. When allowing a ± 1 boundary error for the outputs of the

Stacked Ensemble, the F_1 score went up to 87.9%. Most of these boundary errors on the test set were due to omitting a premodifier or incorrectly including a preceding verb. The first row of Table 6 shows examples of false negatives that fell into this category. The reference annotations appear in **boldface** and the system outputs are surrounded by brackets.

Boundary	Examples
±1	***positive*** [***lymph nodes***] [*repeat* ***the echocardiogram***]
±2	[***overdosing***] ***on insulin*** [***head wound*** *remain dry*] ***1000 ml*** [***fluid restriction***]
Others	***active source of*** [***bleeding***] [*careful monitoring of* ***heart rate***]

Table 6: Examples of boundary errors by the Stacked Ensemble.

When allowing for ±2 boundary word errors, the F_1 score increased to 89.4%. The omission of a prepositional phrase or a pre-modifying phrase and the incorrect inclusion of a verb phrase were frequently observed in these errors. For broader boundaries, the errors are similar to ±2 cases but caused by longer pre-modifying phrases.

We also analyzed false negatives that did not contain any words in common with the outputs of the Stacked Learning Ensemble. For about 34% of the false negative concepts that were missed, none of the words in the concept appeared in the training data.

6 Conclusion

We demonstrated that a Stacked Generalization Ensemble achieves high precision and overall performance comparable to the state-of-the-art for the task of medical concept extraction from clinical notes. Stacked learning offers the advantage of being able to easily incorporate any set of individual concept extraction components because it automatically learns how to combine their predictions to achieve the best performance. We believe that Stacked Generalization offer benefits for many problems in medical informatics because it allows for easy, flexible, and robust integration of multiple component systems, including rule-based systems, external dictionaries and knowledge bases, and machine learning classifiers.

Acknowledgments

This research was supported in part by the National Science Foundation under grant IIS-1018314.

References

Alan R. Aronson and François-Michel Lang. 2010. An overview of MetaMap: historical perspective and recent advances. *J Am Med Inform Assoc*, 17:229–36.

Catherine L. Blake and Christopher J. Merz. 1998. UCI Repository of Machine Learning Databases.

Leo Breiman. 1996. Stacked regressions. *Machine Learning*, 24:49.

Lee M. Christensen, Peter J. Haug, and Marcelo Fiszman. 2002. MPLUS: a probabilistic medical language understanding system. In *Proc. ACL-02 Work. Nat. Lang. Process. Biomed. Domain*. pages 29–36.

Nigel Collier, Chikashi Nobata, and Jun-ichi Tsujii. 2000. Extracting the names of genes and gene products with a hidden markov model. In *Proceedings of the 18th conference on Computational linguistics*. pages 201–207.

Michael Collins. 2002. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on EMNLP*. 1–8.

Berry de Bruijn, Colin Cherry, Svetlana Kiritchenko, Joel Martin, and Xiaodan Zhu. 2011. Machine learned Solutions for Three Stages of Clinical Information Extraction: the State of the Art at i2b2 2010. *J Am Med Inform Assoc*, 18(5):557-562.

Son Doan, Nigel Collier, Hua Xu, Pham Hoang Duy, and Tu Minh Phuong. 2012. Recognition of medication information from discharge summaries using ensembles of classifiers. *BMC Med Inform Decis Mak*, 12:36.

Saso Džeroski and Bernard Ženko. 2004. Is combining classifiers with stacking better than selecting the best one? *Machine Learning*, 54(3):255–273.

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.

Jenny Finkel, Shipra Dingare, Christopher D Manning, Malvina Nissim, Beatrice Alex, and Claire Grover. 2005. Exploring the Boundaries: Gene and Protein Identification in Biomedical Text *BMC Bioinformatics*, 6(S1):S5.

Carol Friedman, Philip O. Alderson, John H. M. Austin, James J. Cimino, and Stephen B. Johnson. 1994. A general natural language text processor for clinical radiology. In *J Am Med Inform Assoc.*, 1(2):161–174

João Gama and Pavel Brazdil. 2000. Cascade generalization. *Machine Learning*, 41(3):315–343.

Peter J. Haug, Spence Koehler, Lee Min Lau, Ping Wang, Roberto Rocha, and Stanley M. Huff. 1995. Experience with a mixed semantic/syntactic parser. In *Proc Annu Symp Comput Appl Med Care*, pages 284–288.

Peter J. Haug, Lee Christensen, Mike Gundersen, Brenda Clemons, Spence Koehler, and Kay Bauer. 1997. A natural language parsing system for encoding admitting diagnoses. In *Proc AMIA Annu Fall Symp*, pages 814–818.

Han-Shen Huang, Yu-Shi Lin, Kuan-Ting Lin, Cheng-Ju Kuo, Yu-Ming Chang, Bo-Hou Yang, I-Fang Chung, and Chun-Nan Hsu. 2007. High-recall gene mention recognition by unification of multiple backward parsing models. In *Proceedings of BioCreative II*, pages 109–111.

Min Jiang, Yukun Chen, Mei Liu, S Trent Rosenbloom, Subramani Mani, Joshua C Denny, and Hua Xu. 2011. A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries. *J Am Med Inform Assoc*, 18(5):601–606.

Ning Kang, Zubair Afzal, Bharat Singh, Erik M. Van Mulligen, and Jan A. Kors. 2012. Using an ensemble system to improve concept extraction from clinical records. *J. of Biomedical Informatics*, 45(3):423–428.

Halil Kilicoglu, Dina Demner-Fushman, Thomas C. Rindflesch, Nancy L. Wilczynski, and R. Brian Haynes. 2009. Towards automatic recognition of scientifically rigorous clinical research evidence. *J Am Med Inform Assoc*, 16: 25–31.

Taku Kudo and Yuji Matsumoto. 2001. Chunking with support vector machines. *Proceedings of NAACL-2001*, pages 1–8.

John D. Lafferty, Andrew McCallum, and Fernando C.N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the International Conference on Machine Learning*. 282–289.

Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proceedings of ACL-2010*. pages 504–513.

Nick Littlestone and Manfred K. Warmuth. 1994. The Weighted Majority Algorithm. *Information and Computation*, 108:212–261.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of ACL-2014: System Demonstrations*. pages 55–60.

Ryan McDonald and Fernando Pereira. 2005. Identifying Gene and Protein Mentions in Text Using Conditional Random Fields. *BMC Bioinformatics*, 6(S1):S6.

M. Netzer, G. Millonig, M. Osl, B. Pfeifer, S. Praun, J. Villinger, W. Vogel, and C. Baumgartner. 2009. A new ensemble-based algorithm for identifying breath gas marker candidates in liver disease using ion molecule reaction mass spectrometry. *Bioinformatics*, 25(7):941–947.

Joakim Nivre and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proc. of ACL 2008*. pages 950–958.

Buzhou Tang, Hongxin Cao, Yonghui Wu, Min Jiang, and Hua Xu. 2013. Recognizing clinical entities in hospital discharge summaries using Structural Support Vector Machines with word representation features. *BMC Med Inform Decis Mak*, 13(S1):S1.

Kay Min Ting and Ian H. Witten. 1999. Issues in stacked generalization, *Journal of Artificial Intelligence Research*, 10:271-289.

Koji Tsukamoto, Yutaka Mitsuishi, and Manabu Sassano. 2002. Learning with multiple stacking for named entity recognition. In *Proc. of COLING-02*. pages 1–4.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *J Am Med Inform Assoc*, 18:552–556.

Shuang-Quan Wang, Jie Yang, and Kuo-Chen Chou. 2006. Using stacked generalization to predict membrane protein types based on pseudo-amino acid composition. *Journal of Theoretical Biology*, 242(4):941–946.

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

Qing T Zeng, Sergey Goryachev, Scott Weiss, Margarita Sordo, Shawn N Murphy, and Ross Lazarus. 2006. Extracting principal diagnosis, co-morbidity and smoking status for asthma research: evaluation of a natural language processing system. *BMC Med Inform Decis Mak*, 6:30.

GuoDong Zhou, Dan Shen, Jie Zhang, Jian Su, and SoonHeng Tan. 2005. Recognition of Protein/Gene Names from Text Using an Ensemble of Classifiers. *BMC Bioinformatics*, 6(S1):S7.

GuoDong Zhou and Jian Su. 2002. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of ACL 2002*, pages 473–480.