Addressing Limited Data for Textual Entailment Across Domains

Chaitanya Shivade* **Preethi Raghavan**† and **Siddharth Patwardhan**†

*Department of Computer Science and Engineering,
The Ohio State University,
Columbus, OH 43210
shivade@cse.ohio-state.edu

†IBM T. J. Watson Research Center,
1101 Kitchawan Road,
Yorktown Heights, NY 10598
{praghav,siddharth}@us.ibm.com

Abstract

We seek to address the lack of labeled data (and high cost of annotation) for textual entailment in some domains. To that end, we first create (for experimental purposes) an entailment dataset for the clinical domain, and a highly competitive supervised entailment system, ENT, that is effective (out of the box) on two domains. We then explore self-training and active learning strategies to address the lack of labeled data. With self-training, we successfully exploit unlabeled data to improve over ENT by 15% F-score on the newswire domain, and 13% F-score on clinical data. On the other hand, our active learning experiments demonstrate that we can match (and even beat) ENT using only 6.6% of the training data in the clinical domain, and only 5.8% of the training data in the newswire domain.

1 Introduction

Textual entailment is the task of automatically determining whether a natural language *hypothesis* can be inferred from a given piece of natural language *text*. The RTE challenges (Bentivogli et al., 2009; Bentivogli et al., 2011) have spurred considerable research in textual entailment over newswire data. This, along with the availability of large-scale datasets labeled with entailment information (Bowman et al., 2015), has resulted in a variety of approaches for textual *entailment recognition*.

A variation of this task, dubbed textual *entailment search*, has been the focus of RTE-5 and subsequent challenges, where the goal is to find all sentences in a corpus that entail a given hypothesis. The mindshare created by those challenges and the availability of the datasets has spurred many creative solutions to this problem. However, the evaluations have been restricted primarily to these datasets, which are in the newswire domain. Thus, much of the existing state-of-the-art research has focused on solutions that are effective in this domain.

It is easy to see though, that entailment search has potential applications in other domains too. For instance, in the clinical domain we imagine entailment search can be applied for clinical trial matching as one example. Inclusion criteria for a clinical trial (for e.g., *patient is a smoker*) become the hypotheses, and the patient's electronic health records are the text for entailment search. Clearly, an effective textual entailment search system could possibly one day fully automate clinical trial matching.

Developing an entailment system that works well in the clinical domain and, thus, automates this matching process, requires lots of labeled data, which is extremely scant in the clinical domain. Generating such a dataset is tedious and costly, primarily because it requires medical domain expertise. Moreover, there are always privacy concerns in releasing such a dataset to the community. Taking this into consideration, we investigate the problem of textual entailment in a low-resource setting.

We begin by creating a dataset in the clinical domain, and a supervised entailment system that

*This work was conducted during an internship at IBM

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 1245–1255,
Berlin, Germany, August 7-12, 2016.

is competitive on multiple domains – newswire as well as clinical. We then present our work on self-training and active learning to address the lack of a large-scale labeled dataset. Our self-training system results in significant gains in performance on clinical (+13% F-score) and on newswire (+15% F-score) data. Further, we show that active learning with uncertainty sampling reduces the number of required annotations for the entailment search task by more than 90% in both domains.

2 Related work

Recognizing Textual Entailment (RTE) shared tasks (Dagan et al., 2013) conducted annually from 2006 up until 2011 have been the primary drivers of textual entailment research in recent years. Initially the task was defined as that of *entailment recognition*. RTE-5 (Bentivogli et al., 2009) then introduced the task of *entailment search* as a pilot. Subsequently, RTE-6 (Bentivogli et al., 2010) and RTE-7 (Bentivogli et al., 2011) featured entailment search as the primary task, but constrained the search space to only those candidate sentences that were first retrieved by Lucene, an open source search engine[1]. Based on the 80% recall from Lucene in RTE-5, the organizers of RTE-6 and RTE-7 deemed this filter to be an appropriate compromise between the size of the search space and the cost and complexity of the human annotation task.

Annotating data for these tasks has remained a challenge since they were defined in the RTE challenges. Successful approaches for entailment (Mirkin et al., 2009; Jia et al., 2010; Tsuchida and Ishikawa, 2011) have relied on annotated data to either train classifiers, or to develop rules for detecting entailing sentences. Operating under the assumption that more labeled data would improve system performance, some researchers have sought to augment their training data with automatically or semi-automatically obtained labeled pairs (Burger and Ferro, 2005; Hickl et al., 2006; Hickl and Bensley, 2007; Zanzotto and Pennacchiotti, 2010; Celikyilmaz et al., 2009).

Burger and Ferro (2005) automatically create an entailment recognition corpus using the news headline and the first paragraph of a news article as near-paraphrases. Their approach has an estimated accuracy of 70% on a held out set of 500 pairs. The primary limitation of the approach is that it only generates positive training examples. Hickl et al. (2006) improves upon this work by including negative examples selected using heuristic rules (e.g., sentences connected by *although*, *otherwise*, and *but*). On RTE-2 their method achieves accuracy improvements of upto 10%. However, Hickl and Bensley (2007) achieves only a 1% accuracy improvement on RTE-3 using the same method, suggesting that it is not always as beneficial.

Recent work by Bowman et al. (2015) describes a method for generating large scale annotated datasets, viz., the Stanford Natural Language Inference (SNLI) Corpus, for the problem of entailment recognition. They use Amazon Mechanical Turk to very inexpensively produce a large entailment annotated data set from image captions.

Zanzotto and Pennacchiotti (2010) create an entailment corpus using Wikipedia data. They hand-annotate original Wikipedia entries, and their associated revisions for entailment recognition. Using a previously published system for RTE (Zanzotto and Moschitti, 2006), they show that their expanded corpus does not result in improvement for RTE-1, RTE-2 or RTE-3.

Similarly, Celikyilmaz et al. (2009) address the lack of labeled data by semi-automatically creating an entailment corpus, which they use within their question answering system. They reuse text-hypothesis pairs from RTE challenges in addition to manually annotated pairs from a newswire corpus (with pairs for annotation obtained through a Lucene search over the corpus).

Note that all of the above research on expanding the labeled data for entailment has focused on *entailment recognition*. Our focus in this paper is on improving *entailment search* by exploiting unlabeled data with self-training and active learning.

3 Datasets

In this section, we describe the data sets from two domains, *newswire* and *clinical*, that we use in the development and evaluation of our work.

3.1 Newswire Domain

For the newswire domain, we use entailment search data from the PASCAL RTE-5, RTE-6 and RTE-7 challenges (Bentivogli et al., 2009; Bentivogli et al., 2010; Bentivogli et al., 2011). The dataset consists of a corpus of news documents, along with a set of hypotheses. The hypotheses come from a separate summarization task, where

[1] http://lucene.apache.org

Dataset	Size	Entailing
Newswire-train	20,104	810 (4.0%)
Newswire-dev	35,927	1,842 (5.1%)
Newswire-test	17,280	800 (4.6%)
Newswire-unlabeled	43,485	-
Clinical-train	7,026	293 (4.1%)
Clinical-dev	8,092	324 (4.0%)
Clinical-test	10,466	596 (5.6%)
Clinical-unlabeled	623,600	-

Table 1: Summary of datasets

```
**NAME[XX (YY) ZZ] has no liver
problems.
PAST MEDICAL HISTORY
1.  Htn
Well controlled
2.  Diabetes mellitus
On regular dose of insulin.

FAMILY HISTORY:
Father with T2DM age unknown
```

Figure 1: Excerpt from a sample clinical note

the summary sentences about a news story (given a topic) were manually created by human annotators. These summary sentences are used as hypotheses in the dataset. Entailment annotations are then provided for a subset of sentences from the document corpus, based on a Lucene filter for each hypothesis.

In this work, we use the RTE-5 development data to train our system (*Newswire-train*), RTE-5 test data for evaluation of our systems (*Newswire-test*), and we use the combined RTE-6 development and test data for our system development and parameter estimation (*Newswire-dev*). We use all of the development and test data from RTE-7, without the human annotation labels, as our unlabeled data (*Newswire-unlabeled*) for self-training and active learning experiments. A summary of the newswire data is shown in Table 1.

3.2 Clinical Domain

There are no public datasets available for textual entailment *search* in the clinical domain. In creating this dataset, we imagine a real-world clinical situation where hypotheses are facts about a patient that a physician seeing the patient might want to learn (e.g., *The patient underwent a surgical procedure within the last three months.*). The unstructured notes in the patients electronic medical record (EMR) is the text against which a system would determine the entailment status of the given hypotheses.

Observe that the aforementioned real-world clinical scenario is very closely related to a question answering problem, where instead of hypotheses a physician may pose natural language questions seeking information about the patient (e.g., *Has this patient undergone a surgical procedure within the past three months?*). Answers to such questions are words, phrases or passages from the patient's EMR. Since we have access to a patient-specific question answering dataset over EMRs[2] (henceforth, referred to as the QA dataset), we use it here as our starting point in constructing the clinical domain textual entailment dataset.

Given a question answering dataset, how might one go about creating a dataset on textual entailment? We follow a methodology similar to that of RTE-1 through RTE-5 for entailment set derived from question answering data. The text corpus in our entailment dataset is the set of de-identified patient records associated with the QA dataset. To generate hypotheses, human annotators converted questions into multiple assertive sentences, which is somewhat similar to what was done in the first five RTE challenges (RTE-1 through RTE-5). For a given question, the human annotators plugged in clinically-plausible answers to convert the question into a statement that may or may not be true about a given patient. Table 2 shows example hypotheses and their source questions. Note that this procedure for hypothesis generation diverges slightly from the RTE procedure, where answers from a question answering system were plugged into the questions to produce assertive sentences.

To generate entailment annotations, we paired a hypothesis with every sentence in a subset of clinical notes of the EHR, and asked human annotators to determine if the note sentence enabled them to conclude an entailment relationship with the hypothesis. For example, the text: *"The appearance is felt to be classic for early MS."* entails the hypothesis: *"She has multiple sclerosis"*. While in the RTE procedure, a Lucene search was used as a filter to limit the number of hypothesis-sentence pairs that are annotated, in our clinical dataset we

[2]a publication describing the question-answering dataset is currently under review at another venue

Question	Hypotheses
When was the patient diagnosed with dermatomyositis?	`The patient was diagnosed with dermatomyositis two years ago.`
Any creatinine elevation?	`Creatinine is elevated.`
	`Creatinine is normal.`
Why were xrays done on the forearm and hand?	`Xrays were done on the forearm and hand for suspected fracture.`

Table 2: Example question → hypotheses mappings

limit the number of annotations by pairing each hypothesis only with sentences from EMR notes containing an answer to the original question in the QA dataset.

The entailment annotations were generated by two medical students with the help of the annotations generated for QA. 11 medical students created our QA dataset of 5696 questions over 71 patient records, of which 1747 questions have corresponding answers. This was generated intermittently over a period of 11 months. Given the QA dataset, the time taken to generate entailment annotations includes conversion of questions to hypotheses, and annotating entailment. While conversion of questions to hypotheses took approx. 2 hours for 20 questions, generating about 3000 hypothesis and text pairs took approx. 16 hours.

At the end of this process, we had a total of 243 hypotheses annotated against sentences from 380 clinical notes, to generate 25,584 text-hypothesis pairs. We split this into train, development and test sets, summarized in Table 1. Although we have a fairly limited number of labeled text-hypothesis pairs, we do have a large number of patient health records (besides the ones in the annotated set). We generated unlabeled data in the clinical domain, by pairing the hypotheses from our training data with sentences from a set of randomly sampled subset of health records outside of the annotated data.

Datasets for the textual entailment search task are highly skewed towards the non-entailment class. Note that our clinical data, while smaller in size than the newswire data, maintains a similar class imbalance.

4 Supervised Entailment System

We begin by defining, in this section, our supervised entailment system (called ENT) that is used as the basis of our self-training and active learning experiments. Our system draws upon characteristics and features of systems that have previously been successful in the RTE challenges in the newswire domain. We further enhance this system with new features targeting the clinical domain. The purpose of this section is to demonstrate, through an experimental comparison with other entailment systems, that ENT is competitive on *both* domains, and is a reasonable supervised system to use in our investigations into self-training and active learning.

4.1 System Description

Top systems (Tsuchida and Ishikawa, 2011; Mirkin et al., 2009) in the RTE challenges have used various types of passage matching approaches in combination with machine learning for entailment. We follow along these lines, and design a classifier-based entailment system. For every text-hypothesis pair in the dataset we extract a feature vector representative of that pair. Then, using the training data, we train a classifier to make entailment decisions on unseen examples. In our system, we employ a logistic regression with ridge estimator (the Weka implementation (Hall et al., 2009)), powered by a variety of passage matching features described below.

Underlying many of our passage match features is a more fine-grained notion of "term match". *Term matchers* are a set of algorithms that attempt to match tokens (including multi-word tokens, such as *New York* or *heart attack*) across a pair of passages. One of the simplest examples of these is *exact string matcher*. A token in one text passage that matches exactly, character-for-character, with a token in another text passage would be considered a term match by this simple term matcher. However, these term matchers could be more sophisticated and match pairs of terms that are synonyms, or paraphrases, or

Exact	String match, ignore case
Multi-word	Overlapping terms in multi-word token
Head	String match head of multi-word token
Wikipedia	Wikipedia redirects and disamb. pages
Morphology	Derivational morphology, e.g. *archaeological* → *archaeology*
Date+Time	Match normalized dates and times
Verb resource	Match verbs using WordNet, Moby thesaurus, manual resources
UMLS	Medical concept match using UMLS
Translation	Affix-rule-based translation of medical terms to layman terms

Table 3: ENT term matchers

equivalent to one another according to other criteria. ENT employs a series of term matchers listed in Table 3. Each of these may also produce a confidence score for every match they find. Because we are working with clinical data, we added some medical domain term matchers as well – using UMLS (Bodenreider, 2004) and a rule-based "translator" of medical terms to layman terms[3].

Listed below are all of our features used in the ENT's classifier. Most passage match features aggregate the output of the term matchers along various linguistic dimensions – lexical, syntactic, semantic, and document/passage characteristics.

Lexical: This set includes a feature aggregating *exact string matches* across text-hypothesis, one aggregating *all term matchers*, a feature counting *skip-bigram matches* (using all matchers), a measure of *matched term coverage of text* (ratio of matched terms to unmatched terms). Additionally, we have some medical domain features, viz. *UMLS concept overlap*, and a measure of *UMLS-based similarity* (Shivade et al., 2015; Pedersen et al., 2007) using the UMLS::Similarity tool (McInnes et al., 2009).

Syntactic: Following the lead of several approaches textual entailment (Wang and Zhang, 2009; Mirkin et al., 2009; Kouylekov and Negri, 2010) we have a features measuring the *similarity of parse trees*. Our rule-based syntactic parser (McCord, 1989) produces dependency parses the text-hypothesis pair, whose nodes are aligned using all of the term matchers. The tree match feature is an aggregation of the aligned subgraphs in the tree (somewhat similar to a tree kernel (Moschitti, 2004)).

Semantic: We apply open domain as well as medical entity and relation detectors (Wang et al., 2011; Wang et al., 2012) to the texts, and post features measuring *overlap in detected entities* and *overlap in the detected relations* across the text-hypothesis pair. We also have a rule-based semantic frame detector for a "medical finding" frame (patient presenting with symptom or disease). We post a feature that aggregates matched *elements of detected frames*.

Passage Characteristics: Clinical notes typically have a structure and the content is often organized in sections (e.g. *History of Illness* followed by *Physical Examination* and ending with *Assessment and Plan*). We identified the section in which each note sentence was located and used them as features in the classifier. Clinical notes are also classified into many different categories (e.g., *discharge summary*, *radiology report*, etc.), which we generate features from. We also generate several features capturing the "readability" of the text segments – *parse failure*, *list detector*, *number of verbs*, *word capitalization*, *no punctuation* and *sentence size*. We also have *a measure of passage topic relevance* based on medical concepts in the pair of texts.

4.2 System Performance

To compare effectiveness of ENT on the entailment task, we chose two publicly available systems – EDITS and TIE – for comparison. Both these system are available under the Excitement Open Platform (EOP), an initiative (Magnini et al., 2014) to make tools for textual entailment freely available[4] to the NLP community. EDITS (Edit Distance Textual Entailment Suite) by Kouylekov and Negri (2010) is an open source textual entailment system that uses a set of rules and resources to perform "edit" operations on the text to convert it into the hypothesis. There are costs associated with the operations, and an overall cost is computed for the text-hypothesis pair, which determines the decision for that pair. This system has placed third (out of eight teams) in RTE-5, and seventh (out of thirteen teams) in RTE-7. The Textual Inference Engine (TIE) (Wang and Zhang, 2009) is a maximum entropy based entailment system relying on predicate argument structure matching. While this system did not partici-

[3]Rules for medical term translator were derived from http://www.globalrph.com/medterm.htm

[4]http://hltfbk.github.io/Excitement-Open-Platform/

System	Newswire			Clinical		
	Precision	Recall	F-score	Precision	Recall	F-score
Lucene	0.47	0.48	**0.47***	0.16	0.22	0.19
EDITS	0.22	0.57	0.32	0.23	0.21	0.20
TIE	0.66	0.21	0.31	0.43	0.01	0.02
ENT	0.77	0.26	0.39	0.42	0.15	**0.23***

Table 4: System performance on test data (* indicates statistical significance)

pate in the RTE challenges, it has been shown to be effective on the RTE datasets. In our experiments, we trained the EDITS system optimizing for F-score (the default optimization criterion is accuracy) and TIE with its default settings. We also used a Lucene baseline similar to the one used in RTE-5, RTE-6 and RTE-7 entailment challenges.

We trained the systems on the training set of each domain and tested on the test set. The Lucene baseline considers the first N sentences (where N is 5, 10, 15 or 20) top-ranked by the search engine to be entailing the hypothesis. The configuration with the top 10 sentences performed the best, and is reported in the results. Note that this baseline is a strong one, and none of the systems participating in RTE-5 could beat it.

Table 4 summarizes the system performance on newswire and clinical data. We observe that systems that did well on RTE datasets, were mediocre on the clinical dataset. We did not, however, put any effort into adaption of TIE and EDITS to the clinical data. So the mediocre performance on clinical is understandable. It is interesting to see though that ENT did well (comparatively) on both domains.

We note that our problem setting is most similar to the RTE-5 entailment search task. Of the 20 runs across eight teams that participated in RTE-5, the median F-Score was 0.30 and the best system (Mirkin et al., 2009) achieved an F-Score of 0.46. EDITS and TIE perform slightly above the median and ENT (with 0.39 F-score) would have ranked third in the challenge.

The performance of all systems on the clinical data is noticeably low as compared to the newswire data. An obvious difference in the two domains is the training data size (see Table 1). However, obtaining annotations for textual entailment search is expensive, particularly in the clinical domain. The remaining sections present our investigations into self-training and active learning, to overcome the lack of training data.

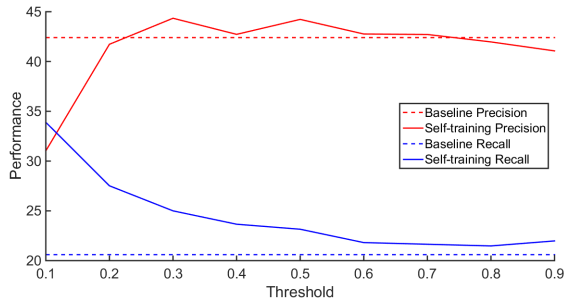
5 Self-Training

Our goal is to exploit unlabeled data, with the hope of augmenting the limited annotated data in a given domain. Self-training is a method that has been successfully used to address limited training data on many NLP tasks, such as parsing (McClosky et al., 2006), information extraction (Huang and Riloff, 2012; Patwardhan and Riloff, 2007), word sense disambiguation (Mihalcea, 2004), etc. Self-training iteratively increases the size of the training set, by automatically assigning labels to unlabeled examples, using a model trained in a previous iteration of the self-training regime.

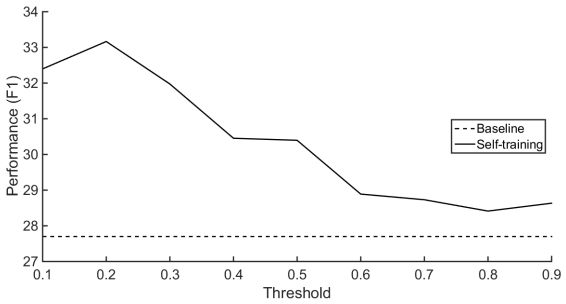
For our newswire and clinical datasets, using the set of unlabeled text-hypothesis pairs U, we ran the following training regime: A model was created using the training data L_n, and applied it to the unlabeled data U. From U, all such pairs that were classified by the model as entailing pairs with high confidence (above a threshold τ) were added to the labeled training data L_n to generate L_{n+1}. Non-entailing pairs were ignored. A new model is trained on data L_{n+1}, and the above process repeated iteratively, until a stopping criteria is reached (in our case, all pairs from U are exhausted).

The threshold τ determines the confidence of our model for a text-hypothesis pair being classified to the entailment class. This threshold was tuned by varying it incrementally from 0.1 to 0.9 in steps of 0.1. The best τ was determined on the development set, and chosen for the self-training system. Figure 2 shows the effect of τ on the development data.

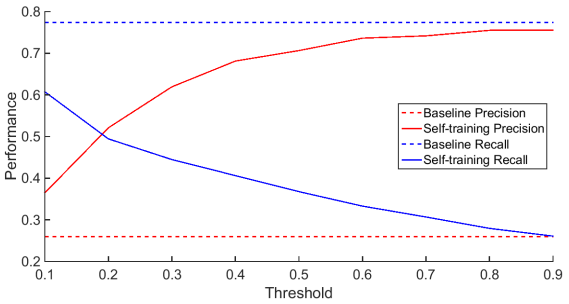
As such, we see that the F-score of the self-trained model is always above that of the baseline ENT system. The F-score increases upto a peak of 0.33 at threshold τ of 0.2 before dropping at higher thresholds. Using this tuned threshold on test set, the comparitive performance on the test set is outlined in Table 5. We observe an F-score



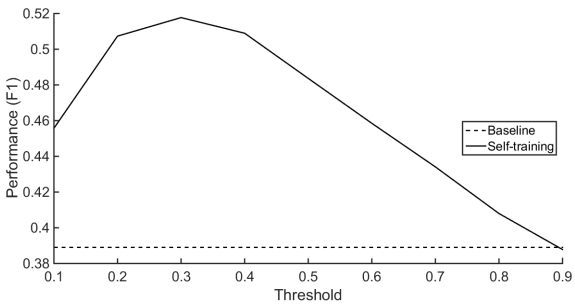
(a) Precision and recall for clinical data



(b) F-Score for clinical data



(c) Precision and recall for newswire data



(d) F-Score for newswire data

Figure 2: Self-training on development data

System	Newswire			Clinical		
	Precision	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
ENT	0.77	0.26	0.39	0.42	0.15	0.23
ENT + Self-Training	0.62	0.48	**0.54***	0.34	0.39	**0.36***

Table 5: Self-training results on test data (* indicates statistical significance)

of 0.36, which is significantly greater than that of the vanilla ENT system (0.23).

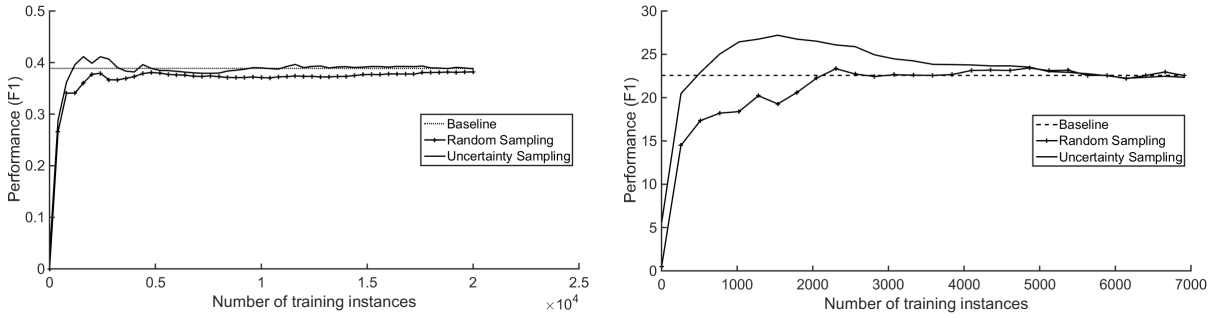
The effect of the threshold on performance correlates with the number of instances added to the training set. When the threshold is low, there are more instances being added (10,799 at threshold of 0.1) into the training set. Therefore, recall is likely to benefit, since the model is exposed to a larger variety of text-hypothesis pairs. However, the precision is low since noisy pairs are likely to be added. When the threshold is high, fewer instances are added (316 at threshold of 0.9). These are the ones that the model is most certain about, suggesting that these are likely to be less noisy. Therefore, the precision is comparatively high.

We also ran our self-training approach on the Newswire datasets. We observed similar variations in performance with newswire data as with the clinical data. At threshold of 0.9, fewer instances (49) are added to the training set from the unlabeled data, while a large number of instances (2,861) are added at a lower threshold τ of 0.1. The best performance (F-score of 0.52) was obtained at threshold of 0.3, on the development set.

This threshold also resulted in the best performance (0.54) on the test set. Similar to the clinical domain, precision increased but recall decreased as the threshold increased. Again, it is evident from Table 5 that gains obtained from self-training are due to recall. It should be noted that the self-trained system achieves an F-score of 0.54 – substantially better than the best performing system of Mirkin et al. (2009) (F-score, 0.46) in RTE-5.

6 Active Learning

Active learning is a popular training paradigm in machine learning (Settles, 2012) where a learning agent interacts with its environment in acquiring a training set, rather than passively receiving independent samples from an underlying distribution. This is especially pertinent in the clinical domain, where input from a medical professional should be sought only when really necessary, because of the high cost of such input. The purpose of exploring



(a) Newswire Data

(b) Clinical Data

Figure 3: Learning curves for uncertainty sampling and random sampling on test data

this paradigm is to achieve the best possible generalization performance at the lowest cost.

Active learning is an iterative process, and typically works as follows: a model M is trained using a minimal training dataset L. A query framework is used to identify an instance from an unlabeled set U that, if added to L, will result in maximum expected benefit. Gold standard annotations are obtained for this instance and added to the original training set L to generate a new training set L'. In the next iteration, a new model M' is trained using L' and used to identify the next most beneficial instance for the training set L'. This is repeated until a stopping criterion is met. This approach is often *simulated* using a training dataset L of reasonable size. The initial model M is created using a subset A of L. Further, instead of querying a large unlabeled set U, the remaining training data $(L - A)$ is treated as an unlabeled dataset and queried for the most beneficial addition.

We carried out active learning in this setting using a querying framework known as *uncertainty sampling* (Lewis and Gale, 1994). Here, the model M trained using A, queries the instances in $(L-A)$ for instance(s) it is least certain for a prediction label. For probabilistic classifiers the most uncertain instance is the one where posterior probability for a given class is nearest to 0.5. To estimate the effectiveness of this framework, it is always compared with a *random sampling* framework, where random instances from the training data are incrementally added to the model.

Starting with a model trained using a single randomly chosen instance, we carried out active learning using uncertainty sampling, adding one instance at a time. After the addition of each instance, the model was retrained and tested on a held out set. To minimize the effect of randomization associated with the first instance, we repeated the experiment ten times and averaged the performance scores across the ten runs.

Following previous work (Settles and Craven, 2008; Reichart et al., 2008) we evaluate active learning using learning curves on the test set. Figure 3 shows the learning curves for newswire and clinical data.

On clinical data, uncertainty sampling achieves a performance equal to the baseline ENT with only 470 instances. With random sampling, over 2,200 instances are required. The active learner matches the performance of the ENT with only 6.6% of training data. Newswire shows a similar trend, with both sampling strategies outperforming ENT, using less than half the training, and uncertainty sampling learning faster than random. While uncertainty sampling matches ENT F-score with only 1,169 instances, random sampling requires 2,305. Here, the active learner matches ENT performance using only 5.8% of the training data.

7 Effect of Class Distribution

After analyzing our experimental results, we considered that one possible explanation for the improvements over baseline ENT could plausibly be because of changes in the class distribution. From Table 1, we observe that the distribution of classes in both domains is highly skewed (only 4-5% positive instances). Self-training and active learning dramatically change the class distribution in training. To assess the effect of class distribution changes on performance, we ran additional experiments, described here.

We first investigated sub-sampling (Japkowicz, 2000) the training data to address class imbalance. This includes down-sampling the majority class or up-sampling the minority class until the classes are

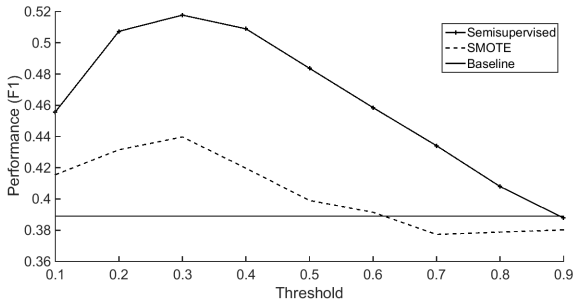


Figure 4: Comparison of SMOTE and self-training (on newswire development set)

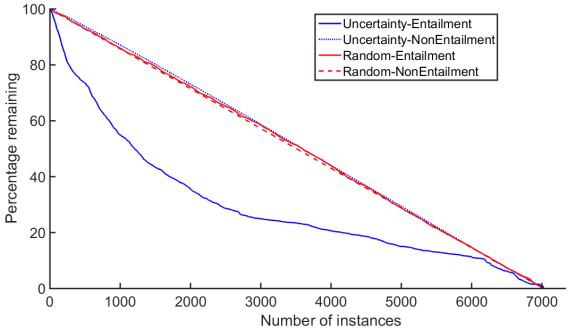


Figure 5: Comparison of sampling strategies for active learning (on newswire development set)

balanced. We found no significant gains over the vanilla ENT baseline with both strategies. Specifically, down-sampling resulted in gains of only 0.002 and 0.001 F-score and up-sampling resulted in a drop of 0.011 and 0.013 F-score on clinical-dev and newswire-dev, respectively.

Another approach to addressing class imbalance is to apply Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002). SMOTE creates instances of the minority class by taking a minority class sample and introducing synthetic examples between its k nearest neighbors. Using SMOTE on newswire and clinical datasets resulted in improvements over baseline ENT in both domains. The improvements using self-training, however, are significantly higher than SMOTE. Figure 4 shows a comparison of SMOTE and self-training on newswire data, where equal number of instances are added to the training set by both techniques.

Finally, for active learning, we consider random sampling as a competing approach to uncertainty sampling. Figure 5 illustrates the percentage of positive and negative instances that get included in the training set for both sampling strategies, as active learning proceeds. The blue solid line shows that positive instances are *consumed* faster than the negative instances with uncertainty sampling. Thus, a higher percentage of positive instances (that approximately equals the number of negative instances getting added) get added and this helps maintain a balanced class distribution.

Once the positive instances are exhausted, more negative instances are added, resulting in some class imbalance that hurts performance (even though more training data is being added overall). In contrast, random sampling does not change the class balance, as it *consumes* a proportional number of positive and negative instances (resulting in more negative than positive instances). The plot indicates that when using uncertainty sampling 80% of the positive examples are added to the training set with less than 50% of the data. This also explains how the active learner matches the performance of the model using the entire labeled set, but with fewer training examples.

8 Conclusion

We explored the problem of textual entailment search in two domains – newswire and clinical – and focused a spotlight on the cost of obtaining labeled data in certain domains. In the process, we first created an entailment dataset for the clinical domain, and a highly competitive supervised entailment system, called ENT, which is effective (out of the box) on two domains. We then explored two strategies – self-training and active learning – to address the lack of labeled data, and observed some interesting results. Our self-training system substantially improved over ENT, achieving an F-score gain of 15% on newswire and 13% on clinical, using only additional unlabeled data. On the other hand, our active learning experiments demonstrated that we could match (and even beat) the baseline ENT system with only 6.6% of the training data in the clinical domain, and only 5.8% of the training data in the newswire domain.

Acknowledgments

We thank our in-house medical expert, Jennifer Liang, for guidance on the data annotation task, our medical annotators for annotating clinical data for us, and Murthy Devarakonda for valuable insights during the project. We also thank Eric Fosler-Lussier and Albert M. Lai for their help in conceptualizing this work.

References

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The Fifth PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the Second Text Analysis Conference*, Gaithersburg, MD.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2010. The Sixth PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the Third Text Analysis Conference*, Gaithersburg, MD.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2011. The Seventh PASCAL Recognizing Textual Entailment Challenge. In *Proceedings of the Fourth Text Analysis Conference*, Gaithersburg, MD.

Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): Integrating Biomedical Terminology. *Nucleic Acids Research*, 32(Database Issue):D267–D270.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal.

John Burger and Lisa Ferro. 2005. Generating an Entailment Corpus from News Headlines. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 49–54, Ann Arbor, MI.

Asli Celikyilmaz, Marcus Thint, and Zhiheng Huang. 2009. A Graph-based Semi-Supervised Learning for Question-Answering. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 719–727, Singapore.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. Recognizing Textual Entailment: Models and Applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–220.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software. *ACM SIGKDD Explorations Newsletter*, 11(1):10.

Andrew Hickl and Jeremy Bensley. 2007. A Discourse Commitment-based Framework for Recognizing Textual Entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 171–176, Prague, Czech Republic.

Andrew Hickl, Jeremy Bensley, John Williams, Kirk Roberts, Bryan Rink, and Ying Shi. 2006. Recognizing Textual Entailment with LCC's GROUNDHOG System. In *The Second PASCAL Recognizing Textual Entailment Challenge: Proceedings of the Challenges Workshop*, Venice, Italy.

Ruihong Huang and Ellen Riloff. 2012. Bootstrapped Training of Event Extraction Classifiers. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 286–295, Avignon, France.

Nathalie Japkowicz. 2000. Learning from Imbalanced Data Sets: A Comparison of Various Strategies. In *Proceedings of the AAAI Workshop on Learning from Imbalanced Data Sets*, pages 10–15, Austin, TX.

Houping Jia, Xiaojiang Huang, Tengfei Ma, Xiaojun Wan, and Jianguo Xiao. 2010. PKUTM Participation at TAC 2010 RTE and Summarization Track. In *Proceedings of the Third Text Analysis Conference*, Gaithersburg, MD.

Milen Kouylekov and Matteo Negri. 2010. An Open-Source Package for Recognizing Textual Entailment. In *Proceedings of the ACL 2010 System Demonstrations*, Uppsala, Sweden.

David D. Lewis and William A. Gale. 1994. A Sequential Algorithm for Training Text Classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, Dublin, Ireland.

Bernardo Magnini, Roberto Zanoli, Ido Dagan, Kathrin Eichler, Neumann Guenter, Tae-Gil Noh, Sebastian Padó, Asher Stern, and Omer Levy. 2014. The Excitement Open Platform for Textual Inferences. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 43–48, Baltimore, MD.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective Self-Training for Parsing. In *Proceedings of Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 152–159, New York City, NY.

Michael McCord. 1989. Slot Grammar: A System for Simpler Construction of Practical Natural Language Grammars. In *Proceedings of the International Symposium on Natural Language and Logic*, pages 118–145, Hamburg, Germany.

Bridget T McInnes, Ted Pedersen, and Serguei V. S. Pakhomov. 2009. UMLS-Interface and UMLS-Similarity : Open Source Software for Measuring Paths and Semantic Similarity. In *Proceedings of the Annual Symposium of the American Medical Informatics Association*, San Francisco, CA.

Rada Mihalcea. 2004. Co-Training and Self-Training for Word Sense Disambiguation. In *Proceedings of the Eighth Conference on Natural Language Learning*, pages 33–40, Boston, MA.

Shachar Mirkin, Roy Bar-Haim, Jonathan Berant, Ido Dagan, Eyal Shnarch, Asher Stern, and Idan Szpektor. 2009. Addressing Discourse and Document Structure in the RTE Search Task. In *Proceedings of the Second Text Analysis Conference*, Gaithersburg, MD.

Alessandro Moschitti. 2004. A Study on Convolution Kernels for Shallow Statistic Parsing. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 335–342, Barcelona, Spain.

Siddharth Patwardhan and Ellen Riloff. 2007. Effective Information Extraction with Semantic Affinity Patterns and Relevant Regions. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 717–727, Prague, Czech Republic.

Ted Pedersen, Serguei V. S. Pakhomov, Siddharth Patwardhan, and Christopher G Chute. 2007. Measures of Semantic Similarity and Relatedness in the Biomedical Domain. *Journal of Biomedical Informatics*, 40(3):288–99.

Roi Reichart, Katrin Tomanek, Udo Hahn, and Ari Rappoport. 2008. Multi-Task Active Learning for Linguistic Annotations. In *Proceedings of ACL-08: HLT*, pages 861–869, Columbus, OH.

Burr Settles and Mark Craven. 2008. An Analysis of Active Learning Strategies for Sequence Labeling Tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079, Honolulu, HI.

Burr Settles. 2012. Active Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114.

Chaitanya Shivade, Courtney Hebert, Marcelo Loptegui, Marie-Catherine de Marneffe, Eric Fosler-Lussier, and Albert M. Lai. 2015. Textual Inference for Eligibility Criteria Resolution in Clinical trials. *Journal of Biomedical Informatics*, 58:S211–S218.

Masaaki Tsuchida and Kai Ishikawa. 2011. IKOMA at TAC2011 : A Method for Recognizing Textual Entailment using Lexical-level and Sentence Structure-level Features. In *Proceedings of the Fourth Text Analysis Conference*, Gaithersburg, MD.

Rui Wang and Yi Zhang. 2009. Recognizing Textual Relatedness with Predicate-Argument Structures. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 784–792, Singapore.

Chang Wang, James Fan, Aditya Kalyanpur, and David Gondek. 2011. Relation Extraction with Relation Topics. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1426–1436, Edinburgh, UK.

Chang Wang, Aditya Kalyanpur, James Fan, Branimir K. Boguraev, and David Gondek. 2012. Relation Extraction and Scoring in DeepQA. *IBM Journal of Research and Development*, 56(3.4):9:1–9:12.

Fabio M. Zanzotto and Alessandro Moschitti. 2006. Automatic Learning of Textual Entailments with Cross-Pair Similarities. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 401–408, Sydney, Australia.

Fabio M. Zanzotto and Marco Pennacchiotti. 2010. Expanding Textual Entailment Corpora from Wikipedia using Co-training. In *Proceedings of the 2nd Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 28–36, Beijing, China.