Mapping Text Phrases to Complex Logical Forms for Semantic Parsing: Exploring Named Entity Recognition

Ashish Gupta ashgup@stanford.edu
Siddharth Jain sjain2@stanford.edu
Sushobhan Nayak nayaks@stanford.edu

CS229: Machine Learning Project
Computer Science Department, Stanford University

1 Introduction

Semantic parsing is the task of matching natural language utterances to formal language representations. Annotated logical forms are usually expensive to acquire and semantic parse, as shown in SEMPRE [2], which learns through question-answer pairs. Question answering systems build a coarse mapping from phrases to predicates using a knowledge base and a large text corpus. They use a bridging operation to generate additional predicates based on neighboring predicates. An example of this is Quora, which is a popular question answering website. Questions asked on the website often lack structure and recommending similar questions requires identifying named entities in agreement with the semantic category expected by a given question.

NER is the task of locating and classifying atomic elements in text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. In SEMPRE [2], NER forms a large part of the algorithm to align the question. Named entity recognition has been studied mostly for information extraction and it's use case has not been assessed in a question answering environment. Questions have different inherent structures than free text, which makes the task non-trivial. We performed some initial experimentation and made the following observations:

- We began our analysis by testing questions on SEMPRE [2]. Our experiments with the QA system showed that due to poor NER, a lot of candidate phrases were missed which led to a drop in performance.
- The system failed to recognize some popular entities (recognized in statements) when presented in the form of questions. Upon looking at the NER model used by the system independently, we found that the NER model performed poorly on questions since it had been trained on statements rather than questions.
- We found the system also performed poorly on unstructured questions. We claim this is attributed to the NER model being trained on well-formed sentences, typically news datasets like Reuters.
- Questions involving certain entities like Movies, Books, TV Shows failed to return answers despite such entities being present in the rules and answer database. On analysis, we found that the NER employed by the system failed to recognize these entities which resulted in the answer not being found.

The above observations motivated us to work on improving Named Entity Recognition specifically for the task of questions and answers. The main contributions of this paper are:

1. Manually annotated a web questions dataset.
2. Scraped IMDB dataset to improve NER for movies.
3. Analyzed and improved NER for QA by varying training sets and models.

The structure of the report is as follows: Section 2 discusses related works, in relation to our paper. Section 3 explains the datasets used to explore NER. In section 4, we list the the models and tools used in classification of named entities. In Section 5, we provide the result and a detailed error analysis of the different models and training sets used. We finally conclude and discuss future work in Section 6.

2 Related Work

SEMPRE [2] uses Lambda Dependency Based Compositional Semantics to represent latent logical forms. If the knowledge base K is treated as a directed graph in which entities are nodes and properties are labels on the edges, then simple Lambda-DCS unary logical forms are tree-like graph patterns with a subset of the tree nodes. SEMPRE recursively constructs distributions over all possible derivations by using i) lexicon mapping from natural language phrases to knowledge base predicates and ii) a set of composition rules.

We focused on the lexicon construction phase where a lexicon is constructed by aligning a large text corpus to a knowledge base (eg. a phrase and predicate aligns if they co-occur with large number of same entities). We evaluated the current candidate phrase generation techniques and performed experiments with the source code of SEMPRE [4] available publicly at http://www-nlp.stanford.edu/software/sempre.

[4] works on named entity recognition in a query setting i.e detection of a named entity in any given query and its corresponding classification. The paper uses Latent Drichlet allocation by considering contexts of named entity as words of a document and classes of the entity as topics. Experimental results indicate that their method can accurately perform named entity recognition in queries.

[6] talks about NER probabilistically by converting multiple entity labels to strings. They compare their QA system with probabilistic NER and traditional QA systems with single entity labels. The paper shows that the added noise introduced by the additional labels is offset by the higher recall gained.

[5] target the identification and classification of named entities in natural language questions. They play with balancing the amount of

free text and questions to optimize the training set for questions. [5] is most closely related to our work. However, one underlying emphasis is the use of unstructured questions in our setting. When querying, users generally tend to forget capitalization and ignore grammar, which we attempt to handle.

3 Dataset

Our main aim was to gather data that was a list of questions. This would allow us to see how Named Entity Recognition worked on single questions. Questions are usually short sentences, that may not be well formed, lack punctuation and may or may not be interrogative in nature. For example, "Who is the president of USA" can also be expressed as "president of USA is" when being searched in a knowledge base.

As we were working with the Stanford NER [2] system, we first gathered the datasets used in this system. Yates compiled a list of 577 questions taken from 81 domains of the Freebase database. The Stanford NER system also referred to another dataset compiled by Berant et al.[2] which consisted of 100 questions. These two datasets were not annotated with any entities. We essentially divided this amongst the team members and manually annotated the datasets. The paper discusses the annotations in this section below.

Mendez et al. [5] performed a similar analysis of named entity recognition on questions. To perform their experiments they had created a dataset of 5,500 annotated questions that was freely available for research. We took their dataset as well and broke it into a Training and Test set for our experiments.

Given that a supervised dataset is not readily available, we figured that a semi-supervised dataset would be good enough for our experiments. As we were trying to also address entity recognition in cultural references like movies, shows and books, we concluded that IMDB would be a good source. IMDB is a web database of movies. The website also publishes a daily webpage which contains news articles on movies. This news webpage has links to other movies/shows, actors/actresses/directors and production houses. We could, therefore, scrape these webpages to generate a dataset of text and use the links to annotate entities. Using a python script we made HTML requests to these webpages over a time period ensuring that the IMDB server does not block our requests (categorizing them as spam). Once we got back the HTML webpage, we massaged the ⟨a href⟩ tags into the appropriate entity tags to create a semi-supervised dataset. As we were relying on the links for tagging entities, not every named entity could be tagged as there were a few actors etc who didn't have links associated with them. We wanted to see what effect this might have on our experiment of annotating web questions.

Finally, we took the annotated Reuters dataset used by Bengio et al. [7] to generate another annotated dataset of around 400,000 words. This dataset was not in the form of questions, but statements. We wanted to show that using statements along with questions as a training set would help induce better performance on a test set of just questions.

All our data is annotated using four named entity tags:

- PER : Denotes the name of a person. Eg. Who is ⟨PER⟩ George Washington ⟨/PER⟩
- LOC : Denotes the name of a location. Eg. What is the capital of ⟨LOC⟩ Egypt ⟨/LOC⟩
- ORG : Denotes the name of an organization. Eg. Who founded ⟨ORG⟩ Google ⟨/ORG⟩
- MISC: Denotes the name of movies, shows, books etc. Eg. Who directed ⟨MISC⟩ the 39 steps ⟨/MISC⟩

For each question and text sentence, we annotate the named entities by marking them between two equal XML tags. Only one category is attributed to each named entity and no nesting of entities as allowed.

Dataset	Tokens	PER	LOC	ORG	MISC
Yates Train	3010	156	65	216	419
Yates Test	1284	55	46	78	108
Berant et al. Test	691	110	36	21	13
Mendez et al. Train	55360	2232	1892	754	0
Mendez et al. Test	3758	86	253	36	0
IMDB Train	41360	1764	0	130	1037
Reuters Train	414863	6813	0	561	4302

Table 1: Dataset Size and Annotations

4 Models and Tools

We used the Stanford NER, MetaOptimize and ETXT2DB software tools available from the web to train our models and evaluate our test dataset. The following are the algorithms used by these tools:

- **Hidden Markov Model (Lingpipe) :** Find parameters to maximize P(X,Y). Assumes features are independent. When labeling Xi future observations are taken into account (forward-backward)
- **Conditional Random Field (Stanford NER) :** Based on Conditional Random Fields, it tries to model the factor graph to generate a discriminative distribution over labels using appropriate features.
- **Support Vector Machine (Minorthird) :** Decision boundaries using features such as Lexical information (Unigram and Bigram), Affix (2-4 suffix and prefix letters), Previous named entities, Possible named entity class, Token feature, Dictionaries (company, person and location names). SVM's main ability is in the inclusion of overlapping features while that of CRF is to include various unrelated features.
- **Perceptron NER (MetaOptimize) :** Augmenting supervised learning with unsupervised word representations as extra features. Also implementing Brown clustering and gazetteers using known lists of named entities.

5 Experiments and Results

5.1 Preliminary Experiments

We drew attention to the notion that the available NER models are usually trained on statements, particularly Reuters dataset which consist of newspaper articles, which have a very higher propertion of statements to questions. To see if it adversely affects the NER task, we ran the Stanford NER [2] (which has been known to outperform most NER models [1]), and various models augmented with gazzeters and word clustering representations from [7]. For Stanford NER,

Algorithm	Train Set	Yates Test Set	Berant et al. Test Set	Mendez et al. Test Set
HMM	Mendez et al.	0.312	**0.707**	0.696
HMM	Yates and Mendez et al.	**0.388**	0.674	**0.702**
SVM	Mendez et al.	**0.452**	**0.539**	**0.752**
SVM	Yates and Mendez et al.	0.418	0.529	0.742
CRF	Mendez et al.	**0.383**	0.528	**0.724**
CRF	Yates and Mendez et al.	0.379	**0.564**	0.721
Stanford NER	Default Model	0.456	**0.723**	0.744
Stanford NER	Default Model (with Capitalization)	0.000	0.000	0.000
Stanford NER	Mendez et al.	0.372	0.588	0.793
Stanford NER	Yates and Mendez et al.	**0.459**	0.659	**0.799**
Stanford NER	IMDB and Yates and Mendez et al.	0.455	0.714	0.796
Stanford NER	Reuters and IMDB and Yates and Mendez et al.	0.411	0.659	0.759
Perceptron NER	Default Model	0.000	0.015	0.007
Perceptron NER	Default Model (with Capitalization)	0.000	0.000	0.000
Perceptron NER	Default Model with Brown Clusters	0.034	0.041	0.039
Perceptron NER	Default Model with Brown Clusters, gazetteers	0.009	0.023	0.037
Perceptron NER	Mendez et al.	0.649	0.892	0.923
Perceptron NER	Yates and Mendez et al.	**0.693**	0.925	**0.962**
Perceptron NER	IMDB and Yates and Mendez et al.	0.681	**0.935**	0.957

Table 2: F1 scores of Named Entity Recognition using different Algorithms

we used an already trained caseless model available from their website, which has been trained on the massive Reuters, MUC and ACE datasets, which gives it robustness against only American or British English. For the other algorithms, we trained the models on a subset of tagged Reuters dataset available from [7]. To investigate the effect of capitalization, we use both caseless and caseful models available for StanfordNER and train the [7] models with capitalization feature switched on and off. The results are shown in Table 2. The F1 scores are phrase F1 scores, i.e., we consider an assignment correct if all the words of the named entity phrase have been assigned a correct tag.[1]

From the table, it's evident that capitalization has a remarkable effect. The trained caseful models perform awfully on our test data, primarily because our test data consists of real questions asked on the web, which are asked with minimal capitalization (e.g. Google search 'who is the author of x' instead of 'Who is the author of X'). This effect is universal: we also tested it with SENNA [3], a state-of-the-art deep learning model, and our F1 score was $\sim$10% (we don't report it in detail here because SENNA in it's readily available form can't be trained, reducing its usefulness in the current investigation where one of our significant contributions is the creation of new datasets). Using a caseless approach takes care of it, but the F1 score is still in the 70's at the most, which is pretty low than the usual numbers in the 80s and 90s for statements [7, 1]. The model fails primarily on ORG and MISC tags. The F1 score for MISC is 0 for both the datasets, denoting a complete failure. Some examples of entities they mispredict are *starry night, 3 juno asteroid, ron glass, the philosopher's stone, six-feet under, x-men, batman: the dark knight returns* etc., which confirms to our conjecture of the model mispredicting names of movies, tv series, books etc. The failure with organization and location detection is attributable primarily to different sentential structure in questions. For example, 'when was interstate 579 formed?' (LOC wrongly tagged as O), 'what is yahoo?' (ORG tagged as O) are wrongly tagged due to lack of precedent structure, while 'what area did the meiji constitution govern?' and 'what was procter & gamble's net profit in 1955?' are tagged with the correct ORG tag due to familiar structures like 'ORG govern' and 'ORG's profit in year'.

[1]The results here might differ from our results in the milestone. We have made significant changes to datasets since then, the starkest of which is including a MISC label along with PER, LOC and ORG we considered in the milestone.

5.2 Augmenting with questions

To combat the lack of questions in the training data, we train our models on the new training set of questions we have prepared - on Mendez and Yates datasets. This is in stark contrast to our previous models; where they were trained on massive news data, our dataset consists of 5500+ tagged questions. But even with such minimal data, the resulting F1 scores are quite close to previous values, and in the case of Yates and Mendez test sets, actually exceeding that of previous models. We have significant improvements in F1s for tags MISC (most of the previously wrongly tagged phrases like *3 juno asteroid, the philosopher's stone* are correctly predicted), LOC and PER when we use both Mendez and Yates datasets for training (see Fig 1 and 2). However, at the same time, we see a dip in the F1 for ORG. Interestingly, the errors are different from the ones we got in the previous run. We make correct predictions for 'what school was delta delta delta founded in?' and 'when was the order of saint michael founded', showing we have begun learning structures particular to questions; at the same time, we make mistakes on 'what area did the meiji constitution govern?' and 'what was procter gamble's net profit in 1955?', pointing to a lack of learning in other sentential structures. It seems that ORG as a tag is more influenced by words around it; most of the sentences with ORG or PER have similar structures and ORG requires specific data points to learn the difference. While it learns question specific structures from this dataset, it loses the statement specific structures it learned from Reuters.

5.3 Including IMDB dataset

5.3.1 Effect of inclusion

We tried nullifying the effect due to lack of questions in the training set in the previous section; we next turn to tackling the issue of mistagging movie and tv series names. To combat that, we have created a huge dataset of IMDB news as described before, and we train our models with a combined dataset of Yates, Mendez and IMDB.

Training Set	Yates Test Set	Berant et al. Test Set	Mendez et al. Test Set
IMDB 1x and Yates and Mendez et al.	0.4545	**0.7135**	**0.7963**
IMDB 5x and Yates and Mendez et al.	**0.466**	0.6871	0.7565
IMDB whole and Yates and Mendez et al.	0.3784	0.6013	0.4426

Table 3: Effect on F1 score when increasing IMDB Training set size

The F1 results don't change much for the Yates test set (0.4545 compared to 0.4589), but we see significant improvement in the Berant et. al. test data, where we achieve the best result of 0.7135 (Table 1). In both cases, we see a significant improvement in tag-F1 for the PER and ORG tags (see Fig 1 and 2). However, at the same time, performance for MISC tag suffers (drops to 0.125 from 0.2 for Yates test set, while remains 0 for Berant). F1 for LOC tag remains more or less the same since the IMDB dataset has no LOC tagged entities. The increase in PER F1 is primarily due to prediction of person entities like 'justin beiber', 'keyshia cole' etc. while the increase in ORG F1 is due to more examples of sentence structures involving ORG in statements (e.g. 'paramount pictures is ...'), which were missing from the question only data bank.

The drop in MISC predictions can primarily be attributed to lack of tags for all the occurrences of movie/tv series names. IMDB dataset is not self-sufficient. The news items that occur in the page have links to corresponding movies, people and studios mentioned in the article. However, not all occurrences are properly linked and as such, a typical example in our dataset looks like this: ***[link]*** *The Hobbit: The Desolation of Smaug* ***[/link]*** *opened at No. 1 in all nine overseas markets in which it debuted on Wednesday,* ***[link]*** *Warner Bros.***[/link]** *reported Thursday. It took in $8.5 million, three percent better than the first film in* ***[link]*** *Peter Jackson* ***[/link]*** *'s trilogy, "An Unexpected Journey," did in its first rollouts last year. France was the top market, as "Smaug" brought in $2.8 million, topping the debut of "Unexpected Journey" by eight percent. Also read: 'Hobbit' Sequel Will Outpace the First at Overseas Box Office – And That's Saying Plenty "The Desolation of Smaug" rolls out Friday in 3,903 U.S. theaters, with midnight screenings in ....* A typical news article repeats the name of a movie multiple times, or mentions related movies, but linking is not done on every instance. On an average, it's the movies that are primarily not linked compared to actors and studios. So while some are tagged as MISC in the training set, a lot of similar structures remain untagged, creating a bias towards un-tagging such occurrences. There are two possible ways to counter this: leverage other specialities of the news articles to improve the tagging (for example, most movies are presented inside brackets) or use a knowledge base of known movies, a gazetteer in particular, to improve the F1 score. We follow the later approach in a future section.

5.3.2 Effect of size

[5] deals with an investigation of a healthy combination of questions and statements for optimal performance. We have already seen that lack of statements leads to poor performance of primarily ORG tags. But lack of enough questions also leads to poor results for F1 on questions. To see if a similar effect is visible in our data, we mixed parts of tagged IMDB data with our tagged question bank. Particularly, we ran three experiments: 1. with equal number of sentences from both IMDB and question bank, 2. with five times the number of sentences from question bank, and 3. the whole of IMDB dateset. The results are in Table 3. We see that the F1 score either decreases or remains almost the same as we increase the number of examples of the IMDB statements. This is natural since as we keep on increasing the number of statements, the effect of structures in the questions are mitigated. In our case, equal number of both seems to be the ideal combination, though a detailed investigation would involve observations at more finer steps, like 1,2,3.. times the size of the question bank. For our purposes, this small investigation is sufficient.

5.4 Introduction of unsupervised information

"If we take an existing supervised NLP system, a simple and general way to improve accuracy is to use unsupervised word representations as extra word features"[7]. To improve upon our current results, we follow this approach of augmenting word feature vectors with unsupervised feature models. In [7], word feature representations involving Brown clusters work the best, so we incorporate that here. The Brown algorithm is a hierarchical clustering algorithm which clusters words to maximize the mutual information of bigrams (Brown et al.,1992). So it is a class-based bigram language model. The hierarchical nature of the clustering means that we can choose the word class at several levels in the hierarchy, which can compensate for poor clusters of a small number of words. One downside of Brown clustering is that it is based solely on bigram statistics, and does not consider word usage in a wider context. Now, Stanford NER is not easily extendable, so we use [7]'s perceptron NER implementation for further experiments.

Another improvement that is usually done in literature is to include prior knowledge of named entities through a gazetteer. Particularly, we leverage a list of entities extracted from Wikipedia and made available by [7]. The list contains names for currencies, places, states, art work, films etc. We augment all of our models with these two features. The final results are shown in Fig 3 and 4. While the default models trained on the Reuters dataset fail miserably, even a small set of questions augmented with Brown clusters and gazetteers outperform them on questions. We have an absolute increase of 25% in F1 scores for both of our test datasets. The final F1 score for the Berant dataset reaches 93%, comparable to results usually achievable on statements, which we set out to achieve.

Another important trend is that inclusion of IMDB dataset improves the F1 scores for the Berant dataset while it brings down the F1 of Yates, a trend seen both when we use Stanford NER and the Perceptron NER with augmented weights. This is because the Berant dataset has more occurrences of questions that involve trivia about Hollywood actors, while Yates doesn't focus too much on them. This is apparent from the individual tag F1 scores: the F1 score for PER remains exactly the same after inclusion of IMDB data, where as it goes up by 2% in case of Berant dataset.

6 Conclusion and Future Work

In this work, we have tried to analyse and improve the existing NER techniques specifically for question answering systems. We began

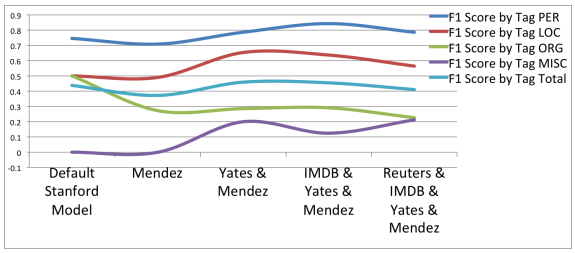

Figure 1: Stanford NER tested on Yates Test set - F1 scores

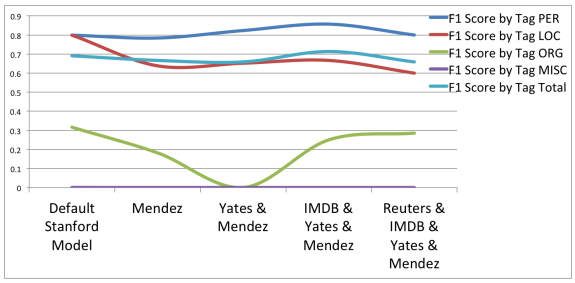

Figure 2: Stanford NER tested on Berant Test set - F1 scores

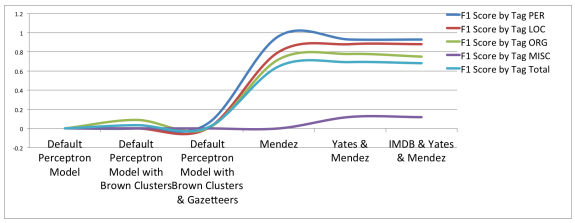

Figure 3: Perceptron NER tested on Yates Test set - F1 scores

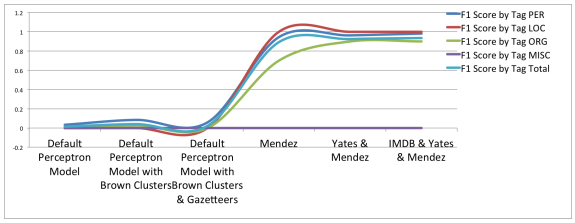

Figure 4: Perceptron NER tested on Berant Test set - F1 scores

with the task of analyzing such systems and quickly realized that NER forms an integral component which was acting as a bottleneck in the performance. Specifically, we looked into issues of the over-dependence of existing models on capitalization features, the lack of questions in training sets for NER models and the effect of external knowledge and domain-specific data, like documents related to movies, on the detection of such entities. Owing to a want of an exhaustive tagged question bank, we created hand-annotated tagged corpora. We also used a semi-supervised approach to create a partially tagged domain specific corpus, a dataset of IMDB news with NER tags, which helped improve our results. Going further, we used unsupervised word cluster features and entity lists to increase the baseline of a state-of-the-art NER system from 78% to 93% on a test set of unstructured and ungrammatical questions typical of a web platform. In future, we would like to integrate the modified named entity recognition models to the question answering system (SEMPRE [2]) and study the performance improvement of the system in detecting answers.

References

[1] Samet Atdag and Vincent Labatut. A comparison of named entity recognition tools applied to biographical texts. *CoRR*, abs/1308.0661, 2013.

[2] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, pages 1533–1544. ACL, 2013.

[3] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, 2011.

[4] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 267–274, New York, NY, USA, 2009. ACM.

[5] Ana Cristina Mendes, Luísa Coheur, and Paula Vaz Lobo. Named entity recognition in questions: Towards a golden collection. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *LREC*. European Language Resources Association, 2010.

[6] Diego Mollá, Menno Zaanen, Steve Cassidy, and North Ryde. Named entity recognition for question answering. In *In Lawrence Cavedon and Ingrid Zukerman, editors, Proceedings of the 2006 Australasian Language Technology Workshop*, pages 51–58, 2006.

[7] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.