Fine-Grained Entity Recognition

Xiao Ling and **Daniel S. Weld**
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350, U.S.A.
{xiaoling, weld}@cs.washington.edu

Abstract

Entity Recognition (ER) is a key component of relation extraction systems and many other natural-language processing applications. Unfortunately, most ER systems are restricted to produce labels from to a small set of entity classes, *e.g.*, *person*, *organization*, *location* or *miscellaneous*. In order to intelligently understand text and extract a wide range of information, it is useful to more precisely determine the semantic classes of entities mentioned in unstructured text. This paper defines a fine-grained set of 112 tags, formulates the tagging problem as multi-class, multi-label classification, describes an unsupervised method for collecting training data, and presents the FIGER implementation. Experiments show that the system accurately predicts the tags for entities. Moreover, it provides useful information for a relation extraction system, increasing the F1 score by 93%. We make FIGER and its data available as a resource for future work.

1 Introduction

Entity Recognition (ER) is a type of information extraction that seeks to identify regions of text (*mentions*) corresponding to entities and to categorize them into a predefined list of types. Entity recognizers have many uses. For example, many *relation extraction* pipelines start by using entity recognition to identify a relation's possible arguments in a sentence and then classify or extract the relation type (Soderland and Lehnert 1994; Banko et al. 2007; Riedel, Yao, and McCallum 2010; Hoffmann et al. 2011). Naturally, the types of the arguments are informative features when determining which relation holds (if any).

Unfortunately, the majority of previous ER research has focused on a limited number of these types. For instance, MUC-7 (Hirschman and Chinchor 1997) considered 3 classes, *person, location* and *organization*. CoNLL-03 added a miscellaneous type (Tjong Kim Sang and De Meulder 2003), and ACE introduced geo-political entities, weapons, vehicles and facilities (Doddington et al. 2004). Other examples include Ontonotes' categorization into 18 classes (Hovy et al. 2006) and BBN's 29 answer types (Weischedel and Brunstein 2005). While these representations are more expressive than the person-location-organization standard, they still fail to distinguish entity classes which are common when extracting hundreds or thousands of different relations (Hoffmann, Zhang, and Weld 2010; Carlson et al. 2010). Furthermore, as one strives for finer granularity, their assumption of mutual exclusion breaks (*e.g.*, Monopoly is both a *game* and a *product*).

There are three challenges impeding the development of a fine-grained entity recognizer: selection of the tag set, creation of training data, and development of a fast and accurate multi-class labeling algorithm. We address the first challenge by curating a set of 112 unique tags based on Freebase types. The second challenge, creating a training sets for these tags, is clearly too large to rely on traditional, manual labeling. Instead, we exploit the anchor links in Wikipedia text to automatically label entity segments with appropriate tags. Next, we use this heuristically-labeled training data to train a conditional random field (CRF) model for segmentation (identifying the boundaries of text that mentions an entity). The final step is assigning tags to the segmented *mentions*; for this purpose, we use an adapted perceptron algorithm for this multi-class multi-label classification problem. We call the complete system FIGER.

In order to evaluate FIGER empirically, we consider two questions: how accurately can it assign tags? And do the fine-grained tags matter? To answer the first question we compare FIGER with two alternative approaches: Stanford's coarse-grained NER system (Finkel, Grenager, and Manning 2005) and Illinois' Named-Entity Linking (NEL, aka Wikifier) system (Ratinov et al. 2011). Although the NEL approach works well for common objects (*e.g.*, Hillary Clinton), our results demonstrate FIGER's advantages on the "long tail" of uncommon entities. To answer the second question, we augment a state-of-the-art relation-extraction system, MultiR (Hoffmann et al. 2011), to accept the types predicted by FIGER as features for the arguments of each potential relation mention. Here we see a 93% boost in F1 score compared to using the Stanford NER tags alone.

In summary, our contributions are multi-fold:

- We introduce a large set of entity types, derived from Freebase, which are expectedly useful both to human understanding and other NLP applications.
- We describe FIGER, a fine-grained entity recognizer,

which identifies references to entities in natural language text and labels them with appropriate tags.

- We compare FIGER with two state-of-the-art baselines, showing that 1) FIGER has excellent overall accuracy and dominates other approaches for uncommon entities, and 2) when used as features, our fine-grained tags can significantly improve the performance of relation extraction by 93% in F1.
- We make the implementation of FIGER and its data available as open source for researchers to use and extend[1].

In the next section we present the design of our system, including tag set curation, generation of heuristically-labeled data and the learning algorithms. We then present our experimental results, discuss related work and conclude with a discussion of future directions.

2 Fine-Grained Entity Recognition

Before describing the whole system, we state the problem at hand. Our task is to uncover the type information of the entity mentions from natural language sentences. Formally speaking, we need to identify the entity mentions $\{m_1, \ldots, m_k\}$, such that each m_i is a subsequence of s,as well as associate the correct entity types, t_i with each m_i.

2.1 Overview

Figure 1 is the overview diagram of our system, FIGER. We divide the whole process into a pipeline. Given a sentence in plain text as input, we first segment the sentence and find the candidates for tagging. Second, we apply a classifier to the identified segments and output their tags. Traditional NER systems (Finkel, Grenager, and Manning 2005) use a sequence model for the whole task, usually a linear-chain Conditional Random Field (CRF) (Lafferty, McCallum, and Pereira 2001). In a sequence model, each token has a corresponding hidden variable indicating its type label. Consecutive tokens with the same type label are treated as one mention with its type. Here the state space of the hidden variables is linear to the size of the type set. However, if one segment is allowed to have multiple labels, the state space will grow exponentially. In practice, this is computationally infeasible when the tag set grows to more than a hundred tags. The pipeline approach avoids this problem and empirically it works reasonably well (Collins and Singer 1999; Elsner, Charniak, and Johnson 2009; Ritter et al. 2011). The models for segmentation and tagging are trained offline.

2.2 Fine-Grained Tag Set

The first step in entity tagging is defining the set of types. While there have been a few efforts at creating a comprehensive tag set (Sekine 2008), no consensus has been reached by the research community. On the other hand, a collabrative knowledge base, such as Freebase, provides thousands of types that are used to annotate each entry/entity in the website[2]. Compared to the type set in (Sekine 2008), the advantages of Freebase types are 1) broader coverage of entities in the world and 2) allowance of an entity bearing multiple overlapping types. For instance, *Clint Eastwood* could be annotated as both *actor* and *director*.

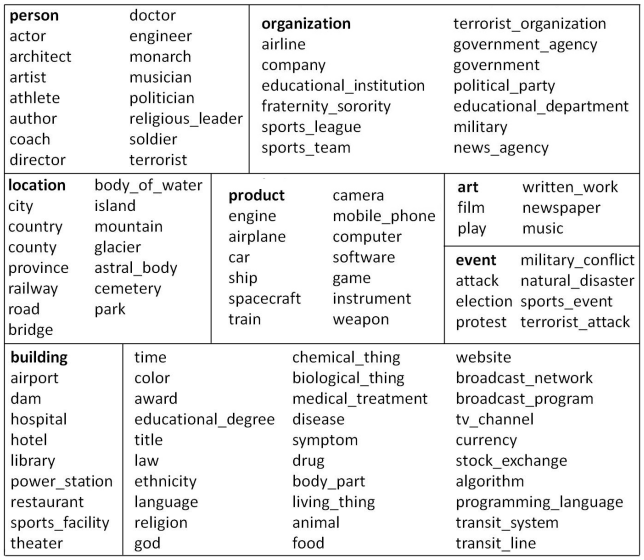

Figure 2: 112 tags used in FIGER. The bold-faced tag is a rough summary of each box. The box at the bottom right corner contains mixed tags that are hard to be categorized.

While Freebase tags are comprehensive, they are also noisy (often created by non-expert users). As a result, we need to filter irrelevant types to reduce the data noise. We only keep well-maintained types (the ones with curated names, *e.g.* `/location/city`) with more than 5 ground instances in Freebase. We further refine the types by manually merging too specific types, *e.g.* `/olympics/olympic_games` and `/soccer/football_world_cup` are merged into *Sports_Event*. In the end, 112 types remain for use as our tag set, denoted as $\mathbf{T}$ (shown in Figure 2).

2.3 Automatically Labeling Data

To effectively learn the tagger, we need a massive amount of labeled data. For this newly defined tag set, there does not exist such a set of labeled data. Previous researchers have hand-labeled each mention in a corpus with the entity types under consideration, but this process is so expensive that only a small training corpus is practical. Instead, we use distant supervision, which is fully automatic and hence scalable (Lengauer et al. 1999). Specifically, we utilize the information encoded in anchor links from Wikipedia text[3] in a manner similar to that of Nothman *et al.* (2008). For each linked segment m in a sentence, we found the corresponding Wikipedia entry e_m via its anchor link, got its types from

[1] http://ai.cs.washington.edu/pubs/310

[2] Wikipedia.com also annotates each article with a set of categories; however, the catogories are too noisy to be effectively used without further processing (Nastase et al. 2010).

[3] We use the Wikipedia dump as of 20110513.

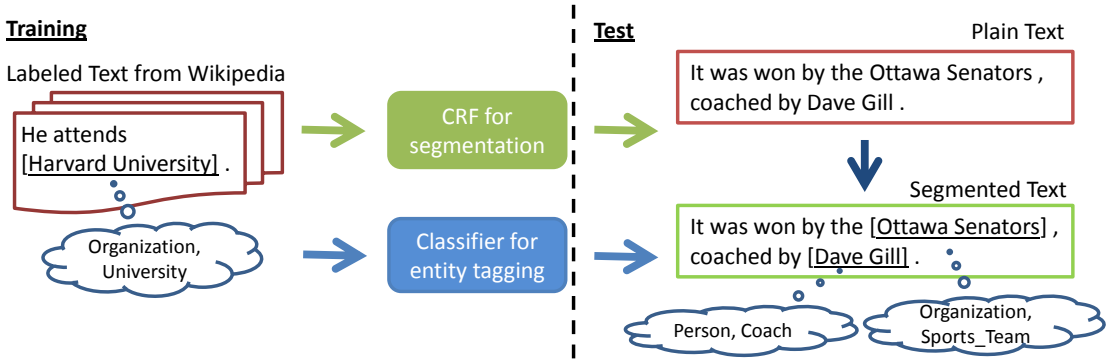

Figure 1: System architecture of FIGER.

Feature	Decription	Example
Tokens	The tokens of the segment.	"Eton"
Word Shape	The word shape of the tokens in the segment.	"Aa" for "Eton" and "A0" for "CS446".
Part-of-Speech tags	The part-of-speech tags of the segment.	"NNP"
Length	The length of the segment.	1
Contextual unigrams	The tokens in a contextual window of the segment.	"victory", "for", "."
Contextual bigrams	The contextual bigrams including the segment.	"victory for", "for Eton" and "Eton ."
Brown clusters	The cluster id of each token in the segment (using the first 4, 8 and 12-bit prefixes).	"4_1110", "8_11100111", etc.
Head of the segment	The head of the segment following the rules by Collins (1999).	"HEAD_Eton"
Dependency	The Stanford syntactic dependency (De Marneffe, MacCartney, and Manning 2006) involving the head of the segment.	"prep_for:seal:dep"
ReVerb patterns	The frequent lexical patterns as meaningful predicates collected in ReVerb.	"seal_victory_for:dep"

Table 1: List of features used in entity tagging. Brown clusters (Brown et al. 1992; Miller, Guinness, and Zamanian 2004) build a partition of words grouped by distributional similarity, which is learned via a probabilistic model from unlabeled text. We used Liang (2005)'s implementation to induce the word clusters. ReVerb (Fader, Soderland, and Etzioni 2011) patterns are mostly multi-word expressions composed of a verb and a noun, with the noun carrying the semantic content of the whole expression. They are complementary to the dependency feature when a single verb is not as meaningful.

Freebase and mapped the original ones into $t_m \subseteq \mathbf{T}$ using the our tag set. We removed the non-sentential sentences by heuristics, *e.g.* thresholding the number of commas and semicolons in a sentence. We also removed the functional pages from Wikipedia, *e.g.* the `List` and `Category` pages. This process therefore automatically annotated sentences from Wikipedia using the tag set $\mathbf{T}$.

2.4 Segmentation

We use a linear-chain CRF model for segmentation[4] with three standard hidden states, *i.e.* "B", "I" and "O". These states indicate, respectively, a beginning token of a mention, a non-beginning token of a mention and a token not in a mention. A maximum sequence of consecutive tokens with "B" as the starting tag and, if any, "I" for the ones after that, is considered an entity mention / segment.

2.5 Multi-class Multi-label Classification

Then FIGER annotates each of the given mentions with a set of types $\hat{t} \subseteq \mathbf{T}$. This tagging problem is characterized in the literature as Multi-class Multi-label Classification (Tsoumakas, Katakis, and Vlahavas 2010). We adapt a classic linear classifier, Perceptron (Rosenblatt 1958) to our problem. A perceptron is in the form of

$$\hat{y} = \arg\max_y w^T \cdot f(x, y)$$

where $\hat{y}$ is a predicted label, $f(x, y)$ is the feature vector of a mention x with a label $y \in \mathbf{T}$ and w is the weight vector of the model. The weights are learned via additive updates

$$w \leftarrow w + \alpha(f(x, y) - f(x, \hat{y}))$$

where y is the true label and $\alpha > 0$ is a parameter controlling the learning pace.

We use all the tags $\hat{t}$ whose scores are larger than zero as the final prediction. And therefore each mention might have more than one predicted tag. We modify the update into

$$w \leftarrow w + \alpha(\sum_{y \in t} f(x, y) - \sum_{\hat{y} \in \hat{t}} f(x, \hat{y}))$$

where t is the set of true tags and $\hat{t}$ is the predicted set. Any spurious mispredictions (i.e. $\hat{t} - t$) will be discouraged. On the other hand, the weights for missed labels (*i.e.* $t - \hat{t}$) will be increased. While learning, the model is trained using heuristically labeled mentions in the text and their tags.

[4]For segmentation, we only use the sentences with all named entities fully labeled.

Features We include various kinds of features we found useful as shown in Table 1. In the table, the segment "Eton" in the sentence "CJ Ottaway scored his celebrated 108 to seal victory for Eton ." is taken as a running example. The Stanford CoreNLP package (Toutanova et al. 2003; Klein and Manning 2003; De Marneffe, MacCartney, and Manning 2006) is applied for syntactic analysis.

3 Experimentation

In this section, we wish to address the following questions:

- How accurately does the system perform on the task of fine-grained entity recognition?
- Are the fine grained entity tags useful for the downstream applications?

3.1 Entity Recognition

First we compare FIGER against two state-of-the-art baselines on the Entity Recognition task.

Data set: From the labeled data set we generated as decribed in Section 2.3, 2 million sentences were randomly sampled for training. We further collected 18 up-to-date news reports[5] We annotated the documents using the tag set T. One entity is allowed to have multiple tags. The annotation was made as complete and precise as possible. In total, 434 sentences were labeled with 562 entities and 771 tags.

Methodology: We use the F1 metric computed from the precision / recall scores in 3 different granualities. All the predictions with wrong segmentation are considered incorrect, which penalizes precision. All labeled entities missed by the system penalize recall. Denote the set of gold segments T and the set of predicted segments P. For a segment e, we denote the true set of tags as t_e and the prediction set $\hat{t}_e$ ($t_e = \emptyset$ if $e \notin T$; $\hat{t}_e = \emptyset$ if $e \notin P$). The three ways of computing precision / recall are listed as follows:

- **Strict:** The prediction is considered correct if and only if $t_e = \hat{t}_e$.

$$precision = (\sum_{e \in P \cap T} \delta(\hat{t}_e = t_e))/|P|\,,$$

$$recall = (\sum_{e \in P \cap T} \delta(\hat{t}_e = t_e))/|T|\,.$$

- **Loose Macro:** The precision and recall scores are computed for each entity.

$$precision = \frac{1}{|P|}\sum_{e \in P} \frac{|\hat{t}_e \cap t_e|}{|\hat{t}_e|}\,,$$

$$recall = \frac{1}{|T|}\sum_{e \in T} \frac{|\hat{t}_e \cap t_e|}{|t_e|}\,.$$

[5]The reports were sampled from the following sources: the student newspaper at University of Washington (dailyuw.com), two local newspapers (adirondackdailyenterprise.com and bozemandailychronicle.com) and two specialized magazines in photography and veterinary (popphoto.com and theveterinarian.com.au). Most entities do not frequently appear in media (*e.g.*students, local residents, etc.) or not exist until very recently (*e.g.*new cameras, experimental drugs, etc.).

Measure	Strict	Loose Macro	Loose Micro
NEL	0.220	0.327	0.381
Stanford (CoNLL)	0.425	0.585	0.548
FIGER	0.471	0.617	0.597
FIGER (GOLD)	0.532	0.699	0.693

Table 2: F1 scores of different systems on entity recognition.

- **Loose Micro:** The overall scores are computed as

$$precision = \frac{\sum_{e \in P} |t_e \cap \hat{t}_e|}{\sum_{e \in P} |\hat{t}_e|}\,,$$

$$recall = \frac{\sum_{e \in T} |t_e \cap \hat{t}_e|}{\sum_{e \in T} |t_e|}\,.$$

Systems Compared: We compare against an adaptation from the Illinois Named-Entity Linking (NEL) system (Ratinov et al. 2011). From the linked results, we look up their oracle types in Freebase, map them into our tag set and use the mapped results as predictions. Note that the mapping process is deterministic and guaranteed correct. We also compare to Stanford NER (Finkel, Grenager, and Manning 2005) using CoNLL (2003)'s 4 classes, i.e. *person*, *organization*, *location* and *miscellaneous*[6]. The perceptron training stops after the 20 iterations with the parameter $\alpha = 0.1$. The values were determined using a seperate validation set.

Results: As seen from Table 2, NEL is not able to identify most entities. The NEL baseline is quite capable in the various linking experiments (Ratinov et al. 2011). Unfortunately, it has the critical disadvantage that its background knowledge base is incomplete and does not contain many of the entities mentioned in everyday news stories. When given a sentence that did contain an entity in Wikipedia, NEL performed extremely well, but this methodology is unlikely to scale to handle the long tail of entities in the world.[7]

Compared to the Stanford NER system, which used a coarser-grained tag set, FIGER successfully discovers more information for the mentioned entities, though its superiority is not dramatic. We were suprised by the relative success of the coarse approach, but attribute it to the fact that 217 of the 562 entities were of type *person* with no subtypes discernible from the text.

We also show the results of FIGER given gold segmentation (the "FIGER (GOLD)" row). The 7%-10% deficit in performance by FIGER with predicted segmentation is partially due to the domain difference from Wikipedia text to newswire.

Another source of errors comes from noise in the training data. For example, "United States" in Freebase is annotated with tags including *language* and *cemetary*. However, not all the training sentences automatically labeled as in Section 2.3 support these types. In other words, there exist false positives in the training data due to the nature of distant supervision. We leave this issue to future work.

[6]for evaluation, each of them is mapped to one of our tag set except *miscellaneous*.

[7]It seems as if the best approach would be to combine an NER system for identifying head entities with FIGERs approach for the long tail; this is an exciting direction for future research.

Relation types		
teamPlaysInLeague (+0.70)	**teamHomeStadium** (+0.10)	**stateHasCapital** (0.00)
stadiumLocatedInCity (+0.53)	**athleteCoach** (+0.09)	**musicArtistGenre** (0.00)
coachesInLeague (+0.44)	**actorStarredInMovie** (+0.06)	**athletePlaysSport** (0.00)
leagueStadiums (+0.35)	**stateLocatedInCountry** (+0.06)	**athleteHomeStadium** (0.00)
cityLocatedInCountry (+0.30)	**radioStationInCity** (+0.05)	**musicianPlaysInstrument** (0.00)
musicianInMusicArtist (+0.24)	**headquarteredIn** (+0.05)	hasOfficeInCountry (-0.01)
cityLocatedInState (+0.23)	**bookWriter** (+0.04)	competesWith (-0.03)
teamPlaysAgainstTeam (+0.20)	**teamPlaysInCity** (+0.02)	televisionStationInCity (-0.05)
coachesTeam (+0.16)	**acquired** (+0.01)	teammate (-0.05)
athletePlaysInLeague (+0.15)	**newspaperInCity** (0.00)	ceoOf (-0.08)
athletePlaysForTeam (+0.12)	**companyEconomicSector** (0.00)	currencyCountry (-0.10)
televisionStationAffiliatedWith (+0.12)	**visualArtistArtMovement** (0.00)	hasOfficeInCity (-0.14)

Table 3: The list of relation types used in the experiment. The number in the brackets following each relation type shows the absolute increase in F1 by MultiR+FIGER over MultiR alone. 29 relation types in bold face show non-negative improvement.

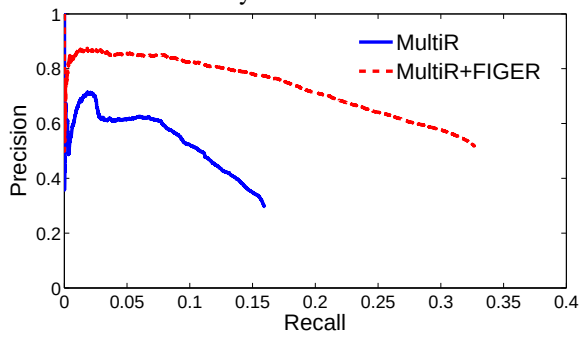

Figure 3: Precision / Recall curves for relation extraction.

3.2 Relation Extraction

We now evaluate FIGER's ability to improve performance at the task of Relation Extraction (RE). We adopted a state-of-the-art RE system, MultiR (Hoffmann et al. 2011) whose source is openly distributed. MultiR is trained using distant supervision by heuristically matching relation instances to text. For example, if $r(e_1, e_2)$=`ceoOf(Steve_Ballmer, Microsoft)` is a relation instance and s is a sentence containing mentions of both e_1 and e_2, then s might be an expression of the ground tuple $r(e_1, e_2)$ and therefore can be easily used as a training example. Unfortunately, the heuristic often leads to noisy data and hence poor extraction quality. MultiR tackles this kind of supervision as a form of multi-instance learning, assuming that there is at least one sentence that naturally supports the fact that $r(e_1, e_2)$ holds[8].

Task: We aim at predicting if $r(e_1, e_2)$ holds given a set of relevant sentences. For testing, we use 36 unique relations proposed by NELL (Carlson et al. 2010)[9]. Another relation *NA* is included when none of the 36 relations (shown in Table 3) holds for a pair of entities.

Data set: We choose the NYT corpus (Sandhaus 2008), which has more than 1.8 million news articles from 1987 to 2007, as the textual repository for matching arguments. All entity pairs present in at least one sentence are considered as a candidate relation instance. The data ordered by date is split into 70% and 30% for training and testing. The features are computed following (Mintz et al. 2009). To get relation labels for these entity pairs, we collect ground tuples for all targeted relations. For each relation r, we start from a set of seed entity pairs which hold for r from NELL database [10]. The set is further enlarged by mapping a relation r_F in Freebase to r and adding all the entity pairs that hold for r_F. These tuples are then used as a gold answer set Δ. The training and test candidates are thus labeled by Δ. If there is no r such that $r(e_1, e_2)$ holds with respect to Δ, the entity pair (e_1, e_2) will then labeled as *NA*.

Methodology: We augment MultiR by allowing it to have FIGER's predictions on the types of the arguments. Binary indicators of FIGER's predicted tags for both arguments (224 in total) are appended to each sentence's feature vector (whose length is in billions) used in MultiR. We compute precision and recall by comparing the collected set of relation instances in the test data, Δ_{test}, and the predicted relation instances by a relation extractor, Δ_{pred}. A true positive is granted if and only if $r(e_1, e_2)$ exists in both Δ_{test} and Δ_{pred}. The precision / recall curve is drawn by varying the confidence thredhold for each relation prediction. Note that the scores are underestimated in that it is likely that a predicted relation instance holds but does not exist in either NELL or Freebase database.

Results: Figure 3 depicts precision / recall curves for two systems, namely the orginal MultiR and the MultiR equipped with FIGER's predictions (MultiR+FIGER). As seen from the curve, FIGER's predictions give MultiR a significant improvement in performance. MultiR+FIGER extended the recall from 15.9% to 32.6% without losing precision. In general, MultiR+FIGER achieved the maximum F1 of 40.0% compared to 20.7% by the original MultiR, showing a 93% increase. Looking more closely at the precision / recall scores on a per-relation basis, we saw that MultiR+FIGER has 22 wins, 7 ties and 7 losses against MultiR alone. Take the most improved relation "teamPlaysInLeague" for example (with a F1 increase from 3.6% to 73%). The type signature for this relation by traditional entity types is at best (*ORG*, *ORG*). This does not make distinction between two arguments. In contrast, FIGER provides (*Sports_team*, *Sports_league*), which obviously exposes key information for the relation extractor.

4 Related Work

In this section, we discuss the previous work on named entity recognition, systems for named entity linking and meth-

[8] We assume binary relations in this experiment. For details of MultiR, we refer interested readers to (Hoffmann et al. 2011).

[9] We excluded the relations having inadequate ground tuples for training.

[10] `http://rtw.ml.cmu.edu/rtw/resources`

ods for multi-class multi-label classification.

Named Entity Recognition (NER) There has been considerable work on NER (Collins and Singer 1999; Tjong Kim Sang and De Meulder 2003; Elsner, Charniak, and Johnson 2009; Ratinov and Roth 2009; Ritter et al. 2011), but this work has several important limitations. Most NER systems only classify into three types: person, location and organization (or miscellaneous). A few systems (*e.g.*, Ontonotes (Hovy et al. 2006)) use more, but still fail to make enough distinctions to provide the most useful features for a relation extraction system.

In the past, there has also been some research focused on building a tagging system using a large number of fine-grained entity types. Fleischman and Hovy (2002) classifies person entities into 8 subcategories. Giuliano and Gliozzo (2008) extends the number to 21 and presents the People Ontology data set, followed by Ekbal *et al.* (2010)'s work enriching the data set by making use of appositional title-entity expressions. Identifying fine-grained person categories indeed is a challenging task but person names only occupy a small portion of all the entities in the world. Sekine (2008) and Lee *et al.* (2006) separately proposed a tag set of around 150 types but there is no implementation publically avaiable. (Nadeau 2007) presented a semi-supervised NER system for 100 types. However, none of these approaches allows overlapping entity types. The exclusion largely reduces the coverage of information embedded in an entity.

Our methods of automatically generating labeled data is inspired by (Nothman, Curran, and Murphy 2008). In contrast, their work is restricted in the traditional tag set while we exploit the Freebase type system and label the Wikipedia text in a much finer-grained tag set $\mathbf{T}$.

Named Entity Linking (NEL) Another highly relevant thread of work is called Named Entity Linking, or Disambiguation to Wikipedia (Bunescu and Pasca 2006; Cucerzan 2007; Milne and Witten 2008; Ferragina and Scaiella 2010; Ratinov et al. 2011). NEL is useful for frequent and well-known named entities. It can be seen as extremely fine-grained entity recognition. The downside is that, as shown in our experiment, it is insufficient when entities do not exist in the background database (e.g. Wikipedia).

Multi-class Multi-label Classification In FIGER, we solved a multi-class multi-label classification problem for tagging the entities. A comprehensive survey on this topic has been written in (Tsoumakas, Katakis, and Vlahavas 2010). We used a simple adaption from the Perceptron model mainly in consideration of speed. With the growing number of labels, a more sophisticated model, *e.g.* (Zhang and Zhou 2007) might achieve a higher accuracy but is highly likely to suffer from unaffordable computational cost.

5 Conclusion

In this paper, we introduce a set of 112 overlapping entity types curated from Freebase types. Secondly, we describe an automatic labeling process by making use of anchor links from Wikipedia text. Third, we present a fine-grained entity recognizer, FIGER, that solves a multi-label multi-class classification problem by adapting a Perceptron model. Our experiments show that FIGER outperforms two other state-of-the-art systems approaches for entity recognition. Most importantly, we demonstrate that a relation extraction system can achieve significant improvement, *i.e.* 93% increase in F1, by using FIGER's predictions as features during extraction.

In the future we hope to explore several directions. We wish to model label correlation to avoid predicting unlikely combinations, *e.g.*, that an entity is both a person and a location. We would like to include non-local features which enforce the consistency of the tags for identical mentions. We also wish to design an approach to reduce the noise stemming from distant supervision. Finally, we hope to show that FIGER can improve the performance of other NLP applications beyond relation extraction.

Acknowledgements The authors thank Congle Zhang for preparing the Relation Extraction part of the experiment and all members of the Turing Center for helpful discussions. We also thank the anonymous reviewers for valuable comments. This material is based upon work supported by a WRF / TJ Cable Professorship, a gift from Google, ONR grant N00014-12-1-0211, and by the Air Force Research Laboratory (AFRL) under prime contract no. FA8750-09-C-0181. Any opinions, ndings, and conclusion or recommendations expressed in this material are those of the author(s) and do not necessarily reect the view of DARPA, AFRL, or the US government.

References

Banko, M.; Cafarella, M.; Soderland, S.; Broadhead, M.; and Etzioni, O. 2007. Open information extraction from the Web. In *Procs. of IJCAI*.

Brown, P.; Desouza, P.; Mercer, R.; Pietra, V.; and Lai, J. 1992. Class-based n-gram models of natural language. *Computational linguistics* 18(4):467–479.

Bunescu, R., and Pasca, M. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of EACL*, volume 6, 9–16.

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Jr., E. R. H.; and Mitchell, T. M. 2010. Toward an architecture for never-ending language learning. In *AAAI*.

Collins, M., and Singer, Y. 1999. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 189–196.

Collins, M. 1999. *Head-driven statistical models for natural language parsing*. Ph.D. Dissertation, University of Pennsylvania.

Cucerzan, S. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of EMNLP-CoNLL*, volume 2007, 708–716.

De Marneffe, M.; MacCartney, B.; and Manning, C. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, 449–454.

Doddington, G.; Mitchell, A.; Przybocki, M.; Ramshaw, L.; Strassel, S.; and Weischedel, R. 2004. The automatic content extraction (ace) program–tasks, data, and evaluation. In *Proceedings of LREC*, volume 4, 837–840.

Ekbal, A.; Sourjikova, E.; Frank, A.; and Ponzetto, S. 2010. Assessing the challenge of fine-grained named entity recognition and classification. In *Proceedings of the 2010 Named Entities Workshop*, 93–101.

Elsner, M.; Charniak, E.; and Johnson, M. 2009. Structured generative models for unsupervised named-entity clustering. In *Proceedings of NAACL-09: HLT*, 164–172. Association for Computational Linguistics.

Fader, A.; Soderland, S.; and Etzioni, O. 2011. Identifying relations for open information extraction. EMNLP.

Ferragina, P., and Scaiella, U. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of CIKM*, 1625–1628. ACM.

Finkel, J.; Grenager, T.; and Manning, C. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of ACL*, 363–370. Association for Computational Linguistics.

Fleischman, M., and Hovy, E. 2002. Fine grained classification of named entities. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*.

Giuliano, C., and Gliozzo, A. 2008. Instance-based ontology population exploiting named-entity substitution. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, 265–272. Association for Computational Linguistics.

Hirschman, L., and Chinchor, N. 1997. Muc-7 named entity task definition. In *Proceedings of the 7th Message Understanding Conference (MUC-7)*.

Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; and Weld, D. S. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*.

Hoffmann, R.; Zhang, C.; and Weld, D. 2010. Learning 5000 relational extractors. In *ACL*.

Hovy, E.; Marcus, M.; Palmer, M.; Ramshaw, L.; and Weischedel, R. 2006. Ontonotes: the 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL*. Association for Computational Linguistics.

Klein, D., and Manning, C. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*.

Lafferty, J.; McCallum, A.; and Pereira, F. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*.

Lee, C.; Hwang, Y.; Oh, H.; Lim, S.; Heo, J.; Lee, C.; Kim, H.; Wang, J.; and Jang, M. 2006. Fine-grained named entity recognition using conditional random fields for question answering. *Information Retrieval Technology* 581–587.

Lengauer, T.; Schneider, R.; Bork, P.; Brutlag, D. L.; Glasgow, J. I.; Mewes, H.-W.; and Zimmer, R., eds. 1999. *Constructing Biological Knowledge Bases by Extracting Information from Text Sources*. AAAI.

Liang, P. 2005. *Semi-supervised learning for natural language*. Master's Thesis, MIT.

Miller, S.; Guinness, J.; and Zamanian, A. 2004. Name tagging with word clusters and discriminative training. In *Proceedings of HLT-NAACL*, volume 4.

Milne, D., and Witten, I. 2008. Learning to link with wikipedia. In *Proceeding of CIKM*, 509–518. ACM.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL*, 1003–1011.

Nadeau, D. 2007. *Semi-Supervised Named Entity Recognition: Learning to Recognize 100 Entity Types with Little Supervision*. Ph.D. Dissertation, University of Ottawa.

Nastase, V.; Strube, M.; Börschinger, B.; Zirn, C.; and Elghafari, A. 2010. Wikinet: A very large scale multi-lingual concept network. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*.

Nothman, J.; Curran, J.; and Murphy, T. 2008. Transforming wikipedia into named entity training data. In *Proceedings of the Australasian Language Technology Association Workshop*, 124–132.

Ratinov, L., and Roth, D. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of CoNLL*, 147–155.

Ratinov, L.; Roth, D.; Downey, D.; and Anderson, M. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proc. of ACL*.

Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of ECML*, 148–163.

Ritter, A.; Clark, S.; Mausam; and Etzioni, O. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of EMNLP*, 1524–1534. Association for Computational Linguistics.

Rosenblatt, F. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review* 65(6):386.

Sandhaus, E. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*.

Sekine, S. 2008. Extended named entity ontology with attribute information. In *Proceeding of LREC*.

Soderland, S., and Lehnert, W. 1994. Wrap-up: a trainable discourse module for information extraction. *J. Artificial Intelligence Research* 2:131–158.

Tjong Kim Sang, E., and De Meulder, F. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of HLT-NAACL*.

Toutanova, K.; Klein, D.; Manning, C.; and Singer, Y. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of NAACL-HLT*. Association for Computational Linguistics.

Tsoumakas, G.; Katakis, I.; and Vlahavas, I. 2010. Mining multi-label data. *Data Mining and Knowledge Discovery Handbook* 667–685.

Weischedel, R., and Brunstein, A. 2005. Bbn pronoun coreference and entity type corpus. Linguistic Data Consortium, Philadelphia.

Zhang, M., and Zhou, Z. 2007. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition* 40(7):2038–2048.