Predicting Accuracy of Extracting Information from Unstructured Text Collections

Eugene Agichtein Silviu Cucerzan
Microsoft Research
One Microsoft Way, Redmond, WA, USA
{eugeneag, silviu}@microsoft.com

ABSTRACT

Exploiting lexical and semantic relationships in large unstructured text collections can significantly enhance managing, integrating, and querying information locked in unstructured text. Most notably, named entities and relations between entities are crucial for effective question answering and other information retrieval and knowledge management tasks. Unfortunately, the success in extracting these relationships can vary for different domains, languages, and document collections. Predicting extraction performance is an important step towards scalable and intelligent knowledge management, information retrieval and information integration. We present a general language modeling method for quantifying the difficulty of information extraction tasks. We demonstrate the viability of our approach by predicting performance of real world information extraction tasks, Named Entity recognition and Relation Extraction.

Categories and Subject Descriptors

H.3.1 [INFORMATION STORAGE AND RETRIEVAL]: Information Search and Retrieval

General Terms Algorithms, Experimentation.

Keywords Language modeling, information extraction, named entity extraction, relation extraction, context language modeling.

1. OVERVIEW

The vast amount of information that exists in unstructured text collections is still primarily accessible via keyword querying at the document level. Unfortunately, this method of information access largely ignores the underlying lexical and semantic relationships between terms and entities in the text. These relationships can be extremely valuable for answering questions, browsing the documents, and managing information associated with the entities of interest. Additionally, document retrieval relevance could be improved if we detect meaningful terms (e.g., named entities such as dates, persons, organizations, and locations); and related entities (e.g., pairs of entities such as "*person*'s birth *date*" and "*person* who invented a *device*") could be used directly to answer questions. Furthermore, indexing entities and relationships would support more intelligent document browsing and navigation, and would allow for richer interactions with the document collections

CIKM'05, October 31-November 5, 2005, Bremen, Germany.

Hence, being able to reliably extract such relationships from text may be of vital importance to knowledge management and information retrieval. However, real collections can exhibit properties that make them difficult for information extraction. At the same time, tuning an information extraction system for a given collection, or porting an information extraction system to a new language, can require significant human and computational effort. Hence, predicting if an extraction task will be successful (i.e., the required information can be extracted with high accuracy) is extremely important for adapting, deploying, and maintaining information extraction systems, and, ultimately, for managing and retrieving information in large text collections.

We observe that document collection properties, such as typical text contexts surrounding the entities or relation tuples, can affect difficulty of an extraction task. In this paper, we present a first general approach to use context language models for predicting whether an extraction task will succeed for a given document collection.

More specifically, we consider two crucial information extraction tasks: *Named Entity Recognition*, and *Relation Extraction*.

- Named Entity Recognition (NER) is a task of identifying entities such as "Person", "Organization", and "Location" in text. The ability to identify such entities has been established as an important pre-processing task in several areas including information extraction, machine translation, information retrieval, and question answering. NER often serves as an important step in the Relation Extraction task described next.
- Relation Extraction (RE), is a task of identifying semantic relationships between entities in the text, such as "*person*'s birth *date*", which relates a person name in the text to a date that is the person's birth date. Once the tuples for this relation (e.g., <"Albert Einstein", "14 March 1879">) are identified, they can be used to directly answer questions such as "When was Albert Einstein born?"

Most state of the art NER and RE systems rely on local context to identify entities or determine the relationship between target entities. In NER, contextual patterns such as "*Mr.*" or "*mayor of*" are often used for hypothesizing occurrences of entities and classifying such identified entities, especially when they are polysemous or of a foreign origin. The local context is also important for the RE task. Intuitively, if the context surrounding

the entities of interest for a given relation looks similar to the general text of the documents (i.e., there are no consistent and obvious "clues" that the entities or relationships of interest are present), then the RE task for that relation will be hard. While NER systems can resort to dictionary lookups in some cases (e.g., for the "Location" entities, dictionaries can be particularly helpful), for others (e.g., people's names or organizations) high accuracy may not be possible. In contrast, if the text context around entities in the collection tends to contain telltale clues, such as "Mr." preceding a person name, the extraction task is expected to be easier, and higher accuracy achievable.

Our approach formalizes and exploits this observation by building two *language models* for the collection – a task-specific *context language model* for the extraction task, and the overall background model for the collection. We can then compare the two language models and compute the divergence of the task-specific language model from that of the overall collection model. If the divergence is high (i.e., the task-specific language model is different from the overall model), the extraction task is expected to be easier than if the divergence is low (i.e., the task-specific language model is very similar to the document language model).

Interestingly, our approach can be potentially helpful for other applications, including better term weighting for information retrieval, and supporting active learning for interactive information extraction. For example, we could derive improved term weighs for domain-specific retrieval tasks such as birthday finding by incorporating context model weights. We will discuss other promising future directions of this work in Section 5.

The rest of this paper is organized as follows. In the next section we review related work. In Section 3 we present our formal model and the algorithms for building the language models. In Section 4 we present experimental results for the NER and RE tasks over large document collections. In Section 5 we present our conclusions, and discuss potential future research directions.

2. RELATED WORK

Our work explores language modeling for information extraction and thus touches on areas of information retrieval, information extraction, and language modeling.

Our approach is partly inspired by the work of Cronen-Townsend, Zhou, and Croft [9] on predicting query performance by measuring the similarity of a language model LM_Q derived from the retrieved documents for a query and a language model for the whole target collection of documents LM_{Coll}. Using simple unigram language models, they showed that the relative entropy between the query and collection language models correlates with the average precision in several TREC test collections. In this paper, we apply similar language modeling techniques to the task of predicting information extraction performance.

Language modeling, typically expressed as the problem of predicting the occurrence of a word in text or speech, has been an active area of research in speech recognition, optical character recognition, context-sensitive spelling, and machine translation. An in-depth analysis of this problem in natural language processing is presented in [20], Chapter 6. Language modeling has also been used to improve term weighting in information retrieval (e.g., [22, 30] and others). However, in previous work LM was used as a tool for improving the specific system performance, whereas in our work we attempt to predict performance for general extraction tasks.

An important distinction of our work is that we consider *task-specific* contexts. As our results indicate, using the locality in the overall document collection may not be sufficient, as local context models can become similar to the background model for overall document collection. Our approach is similar in spirit to the use of entity language models described in [23] for classifying and retrieving named entities. Our work is complementary as we present a general approach for modeling the performance of extraction tasks including both named entity recognition and relation extraction.

For the named entity recognition task, numerous ways of exploiting local context were proposed, from relatively simple character-based models such as [11] and [19] to complex models making use of various lexical, syntactic, morphological, orthographical information, such as [6] and [10]. In this work, we show that we can predict the difficulty of identifying several types of named entities by using relatively simple context language models. This study can be viewed as complementary to Collins' work [8] on the difficulty of identifying named entity boundaries, regardless of entity type.

Relation extraction systems rely on variety of features (e.g., syntactic, semantic, lexical, co-occurrence), but all depend heavily on context. Once the entities are identified, it is the textual context that expresses the relationship between the entities. Partially supervised relation extraction systems (e.g., [1], [12], [18], [24], and others) rely on the text contexts of example facts to derive extraction patterns.

For relation extraction, the task difficulty was previously analyzed by considering the complexity of the target extraction templates ([2] and [17]). Another promising approach described in [13] modeled the task domain variability by considering the different paraphrases used to express the same information in the text. In contrast, our work quantifies the difference between the contexts around the entities and unrelated text contexts. If the contexts of the example facts are similar to the background text, an extraction system is expected to have more difficulty deriving extraction patterns and recognizing the relevant entities.

3. MODELING EXTRACTION DIFFICULTY

In this section we describe the general approach we take for modeling the difficulty of an extraction task, and hence the expected performance of an extraction system on the task (Section 3.1). Then, in Section 3.2, we describe the algorithms for computing the language models to make our predictions.

3.1 Model

As we discussed, the textual context (i.e., the local properties of the text surrounding the entities and relations of interest) can be of crucial importance to extraction accuracy. Intuitively, if the contexts in which the entities occur are similar to the general text then extraction is expected to be difficult. Otherwise, if there are

strong contextual clues, the extraction should be easier and we should expect higher extraction accuracy.

To quantify the notion of context, we use a basic unigram language model, which is essentially a probability distribution over the words in the text's vocabulary. In this study, we derive this probability distribution from the histogram of words occurring in the local context of target entities by using maximum likelihood estimation. Our purpose is to compare the language model associated with an entity type or relationship LM_C with a background language model for the whole target text, denoted by LM_{BG}. Therefore, no smoothing of these models is necessary. Intuitively, if the background language model for the collection is very similar to the language model constructed from the context of the valid entities then the task is expected to be hard. Otherwise (if LM_C is very different from LM_{BG}), the task is expected to be easier.

A common way to measure the difference between two probability distributions is relative entropy, also known as the Kullback-Leibler divergence:

$$\mathrm{KL}(LM_C \,||\, LM_{BG}) = \sum_{w \in V} LM_C(w_i) \cdot \log \frac{LM_C(w)}{LM_{BG}(w)}$$

In Information Theory, KL-divergence represents the average number of bits wasted by encoding messages drawn from the distribution LM_C using as model the distribution LM_{BG}.

Alternatively, we can measure how different two models are by using cosine similarity, which represents the cosine of the angle between the two language models seen as vectors in a multidimensional space in which each dimension corresponds to one word in the vocabulary:

$$\mathrm{Cosine}(LM_C, LM_{BG}) = \frac{< LM_C \cdot LM_{BG} >}{\| LM_{BG} \| \cdot \| LM_C \|}$$

The closer the cosine is to 1, the smaller the angle and thus, the more similar the two models. Hence, to measure the difference of the two models LM_C and LM_{BG} we define *CDist* as:

$$CDist(LM_C \,||\, LM_{BG}) = 1 - Cosine(LM_C, LM_{BG})$$

to maintain symmetry with the *KL* metric, with larger values indicating larger difference between models.

3.2 Constructing the Language Models

We now describe how to construct a language model for a given extraction task. For clarity, we describe a unigram language model, but our methodology can be extended to higher-order features. For syntax-based extraction systems, we could parse the text and incorporate that information into the model as in [6]. However, as we will show experimentally, a simple unigram model is sufficient to make useful predictions.

To construct the task-specific context language model LM_C we search the collection for occurrences of valid entities (or relation tuples). While for the NER and RE tasks LM_C is constructed slightly differently (as described below), the overall approach is to consider the text context to be the *K* words surrounding the entities in question.

More specifically, the language model for NER is constructed as outlined in Figure 3.1. We scan the document collection D, searching for occurrences of each known entity E_i. When an entity is detected, we add to LM_C up to *K* terms to the right and to the left of the entity.

The algorithm for constructing a task-specific language model for RE is outlined in Figure 3.2. The procedure is similar to the NER algorithm above. We scan the document collection D, searching for occurrences of each known example tuple T_i for the target relation. For this, we search for all attributes of T_i in the text. If all entities are present, and occur within *K* words of each other, we increment the LM_C counts of all the words between the leftmost and the rightmost entities. If the entities in a relation tuple are close together (i.e., there are fewer than *K* words separating the entities in the text), we include all the terms separating the entities.

```
ConstructNERLanguageModel (Entities E, Documents D, K )
For each document d in D
    For each entity E_i in E
        if E_i is present in d
            For each instance of E_i spanning from start to end
                For each term w in d [start – K], …, d [start-1]
                    Increment count of w in LM_C
                For each term w in d [end + 1], …, d [end + K]
                    Increment count of w in LM_C
Normalize LM_C
return LM_C
```

Algorithm 3.1: NER Context language model construction.

```
ConstructRELanguageModel (Tuples T, Documents D, K )
For each document d in D
    For each tuple T_i=(t_i^1,t_i^2) in T
        If t_i^1 or t_i^2 not present in d continue
        For each pair of adjacent instances of t_i^1,t_i^2
                occurring at positions start and end
                and separated by fewer than K words
            For each term w in d [start +1],…, d [end - 1]
                Increment count of w in LM_C
Normalize LM_C
return LM_C
```

Algorithm 3.2: RE Context language model construction.

Unfortunately, we don't have all the valid entities available (i.e., when predicting whether a task will succeed without going through the complete extraction process). Hence, our model is build based on *sampling* the collection using a small (20-40) sample of the known entities or tuples by providing only these example entities as input to the NER and RE language model construction algorithms above. For a large corpus, the sample-based model is expected to be a reasonable approximation of the complete task specific language model.

The background language model, LM_{BG} is derived through maximum likelihood estimation using the word frequencies in each document collection. When we discard stopwords from LM_C we also discard them from LM_{BG}.

In order to interpret the divergence of a task specific language model LM_C from the background language model, we build a reference context language model LM_R (also denoted as RANDOM). We construct LM_R, by taking random samples of words in the vocabulary (excluding stopwords) of the same size as the entity samples. We then use these words input to Algorithm 3.1. Using LM_R we can then compute the "reference" divergence of a context language model from the background model for a given sample size. For large sample sizes, LM_R is expected to approximate the background model. Indeed, Figure 3.1 reports that for larger random word sample sizes, LM_R becomes more similar to the background model, and the divergence steadily decreases.

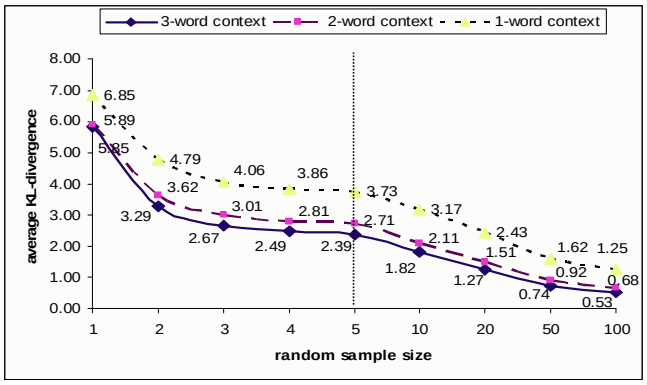

Figure 3.1: The average KL-divergence between the context language models for random samples of words and the background language model.

We also use LM_R to normalize our divergence measures to be robust to different entity sample sizes and collection sizes. For this, we compute the *normalized divergence* as the ratio of the KL-divergence value of LM_C, and the KL-divergence value of LM_R, as compared to the background distribution LM_{BG}. We can similarly compute *normalized cosine distance* as the ratio of the CDIST values of LM_C and LM_R compared to LM_{BG}.

Constructing the context models LM_C and LM_R can be done efficiently by using any off-the-shelf search engine and considering only the documents retrieved by search for the example entities or tuples, and run Algorithms 3.1 and 3.2 only over these reduced document sets. Having described constructing the language models for extraction tasks, we now turn to experimental evaluation.

4. EXPERIMENTS

We evaluated our prediction for two real-world tasks: Named Entity Recognition (NER) and Relation Extraction (RE). We first describe the experimental setup (Section 4.1), including the datasets, entity and relation types, and parameter settings we considered. Then we describe our experiments for predicting NER difficulty (Section 4.2), followed by our experiments on predicting RE difficulty (Section 4.3).

4.1 Experimental Setup

In order to design a realistic evaluation we focused on two extraction tasks, NER and RE, over large document collections. The overall goal of the experiments is to determine if the language models, constructed from a realistically small sample of the extractions of interest, can make useful predictions about the observed accuracy of the extraction task for that collection.

Task	*Collection*	*Size*
NER	Reuters RCV1, 1/100	3,566,125 words
	Reuters RCV1, 1/10	35,639,471 words
	EFE newswire articles, May 2000 (Spanish)	367,589 words
	"De Morgen" articles (Dutch)	268,705 words
RE	Encyclopedia documents	64,187,912 words

Table 4.1: Document collections used in experiments.

The document collections used for these experiments are reported in Table 4.1. The Reuters RCV1 documents were drawn from the collection used in the CoNLL 2003 [17] NER shared task evaluation. The EFE (Spanish) and the De Morgen (Dutch) documents were the datasets used in the CoNLL 2002 NER shared task evaluation. Note that while the Spanish and the Dutch collections are small, they are a "standard" dataset for NER evaluation. For the RE experiments, we used Encarta, a large online encyclopedia document collection.

For all experiments, we start with a small sample (10-40) of entities or relation tuples, drawn at random from a list of known valid entities or tuples. In Table 4.2 we report the specifics of the extraction tasks used for the experiments.

Type of Task	*Sample Extractions (Description)*	*Task*
NER (Named Entity Recognition)	Location names	LOC
	Miscellaneous named entities	MISC
	Organization names	ORG
	Person names	PER
RE (Relation Extraction)	Person's birth dates	BORN
	Person's death dates	DIED
	Person's inventions	INVENT
	Person's writings	WROTE

Table 4.2: Entities and relations used in experiments.

To validate our extraction performance predictions for the NER task, we used as reference the top performing systems in the CoNLL shared task competition, which were evaluated over a manually annotated subset of news articles from the same RCV1 corpus as described above. Moreover, we built the samples of named entities by randomly sampling the set of named entities present in the training set provided by the CoNLL competition organizers (described in [26] and [27]).

To validate our performance predictions on the RE task, we used a bootstrapping-based extraction system similar to Snowball [1], which is heavily dependent on the example entities and the text context in which they appear to derive extraction patterns. For comparison, we also report RANDOM, the divergence of the random keyword sample-based language model, LM_R.

In our experiments we explored the following parameters:

- Context size: number of words to the left and to the right of entity to include as context.

- Maximum distance separating the entities (for RE task)
- Divergence metric, *CDist* or *KL*: The language model divergence metrics defined in Section 3.1. We found that CDIST is strongly correlated with KL, and does not provide additional information. Therefore, for clarity and brevity, we report only the KL values for our experimental evaluation.
- Example set size *S*: number of randomly drawn entities (or relation tuples).
- Random sample size *R*: number of randomly drawn terms to estimate the background model. For all experiments, *R* was equal to the value of *S* above for each task.
- *Stopwords*: we analyze two cases, when stopwords (common English words such as prepositions, conjunctions, numerals, etc.) are included in the language model, and when they are excluded. In both cases, we discard punctuation.
- *N*-gram size *N*: We considered word unigrams, bigrams and trigrams as features of the language models.

4.2 Predicting NER Difficulty

In order to explore the parameter space and evaluate the accuracy of our predictions, we use as reference the reported performance of the top five systems in the CoNLL 2003 shared task competition [27], which is summarized in Table 4.3. According to the reported numbers, the Person (PER) and Location (LOC) entities are the "easiest" to extract, whereas the Miscellaneous (MISC) and Organization (ORG) are relatively difficult.

	Florian et al. [15]	Chieu et al. [7]	Klein et al. [19]	Zhang et al. [31]	Carreras et al. [5]	Average
LOC	91.15	91.12	89.98	89.54	89.26	**90.21**
MISC	80.44	79.16	80.15	75.87	78.54	**78.83**
ORG	84.67	84.32	80.48	80.46	79.41	**81.86**
PER	93.85	93.44	90.72	90.44	88.93	**91.47**
Overall	88.76	88.31	86.31	85.50	85.00	86.77

Table 4.3: F-measures on the Reuters RCV1 collection reported by the top 5 systems participating in the CoNLL 2003 Shared Task competition.

We report the results of our system on predicting NER difficulty in Tables 4.4a, 4.4b, and 4.5. The first two tables present the results obtained on a smaller subset of the Reuters RCV1 corpus, of 3.5 million words, while the latter shows the results obtained for a 10 times bigger subset of the same corpus (35 million words). It is remarkable that the language models estimated on the smaller corpus make extremely similar predictions to those estimated on a corpus 10 times larger.

Our ranking identifies ORG and PER entities as "easy" to extract entity types and LOC and MISC as hard to extract. These correlate with the results reported by the participants in the CoNLL 2003 Shared Task competition (Table 4.2), with the exception of the LOC entities. We believe this happens for three reasons: first, the location entities in the test set overlap to a large degree with the locations in the training data; second: more than for the other entity types considered, indicative contexts of LOC entities are represented by stopwords (e.g. *in*, *from*, *to*), third, all systems shown in Table 4.3 except [19] used extensive lists of gazetteers, which were likely to contain most locations that news articles may talk about and thus, covering most of the locations in the test.

Context Size 1	Sample 1	Sample 2	Sample 3	Average	Normalized
LOC	1.47	1.54	**1.49**	1.50	0.99
MISC	1.31	**2.09**	2.29	1.89	1.25
ORG	4.36	2.25	**4.12**	3.57	2.36
PER	7.40	4.08	**5.28**	5.58	3.69
RANDOM	**1.57**	1.24	1.73	1.51	

Context Size 2	Sample 1	Sample 2	Sample 3	Average	Normalized
LOC	**1.12**	1.15	1.11	1.12	1.21
MISC	0.94	**1.46**	1.62	1.34	1.46
ORG	**3.60**	1.85	3.85	3.10	3.37
PER	5.71	3.22	**4.30**	4.41	4.79
RANDOM	1.04	0.73	**1.00**	0.92	

Context Size 3	Sample 1	Sample 2	Sample 3	Average	Normalized
LOC	0.92	0.94	**0.93**	0.93	1.19
MISC	0.78	**1.18**	1.33	1.09	1.40
ORG	**2.95**	1.65	3.45	2.68	3.44
PER	5.16	2.80	**3.79**	3.91	5.01
RANDOM	0.87	0.63	**0.83**	0.78	

Table 4.4a: Absolute and Normalized KL-divergence of LM_C for varying context sizes for 3 random samples of 20 entities (RCV 1/100, including stopwords).

	Context Size 1		Context Size2		Context Size 3	
	Absolute	Normalized	Abs.	Norm.	Abs.	Norm.
LOC	2.52	1.03	1.78	1.19	1.48	1.17
MISC	3.22	1.33	2.30	1.53	1.83	1.44
ORG	5.27	2.17	4.40	2.93	3.81	3.00
PER	7.64	3.14	6.27	4.18	5.62	4.43
RANDOM	2.43		1.50		1.27	

Table 4.4b: Absolute and normalized KL-divergence of LM_C for varying context sizes for 3 random samples of 20 entities (RCV 1/100, discarding stopwords).

	Context Size 1		Context Size 2		Context Size 3	
	Absolute	Normalized	Abs.	Norm.	Abs.	Norm.
LOC	1.76	0.88	1.20	1.06	0.98	1.07
MISC	2.51	1.26	1.67	1.47	1.29	1.40
ORG	4.25	2.12	3.36	2.95	2.83	3.08
PER	5.88	2.94	4.68	4.11	4.10	4.46
RANDOM	2.00		1.14		0.92	

Table 4.5: Absolute and normalized KL-divergence of LM_C for varying context sizes for 3 random samples of 20 entities (RCV 1/10, discarding stopwords).

Tables 4.4a and 4.4b report the prediction results using stopwords in the language model (Table 4.4a) and discarding the stopwords (Table 4.4b). As expected, the context language models are more similar to the background model when stopwords are included, but in both cases the conclusions are the

same. This is encouraging, as it shows that our approach may work even for languages where no lexical information (such as stopwords) is known *a priori*.

A drawback of our current approach is that we do not consider how easy it is to identify entities of a given type based on sources of information other than context, such as morphology, internal capitalization, or gazetteer lists. Consequently, our system may not be able to predict accurately the extraction performance of fully-featured systems for entities with various intrinsic properties that make them easier or harder to identify independent of context (e.g. the real performance for MISC is somewhat lower than expected based on our prediction due to the fact that capitalization and length varies to a much larger degree for MISC entities than for the other entity types).

We now consider the sensitivity of our results to sample size (Figure 4.1) and N-gram size (Figure 4.2). Figure 4.1 reports normalized KL divergence for RCV 1/100 for seed sample sizes of 10, 20, 30, 40, and 50. As we can see, for sample sizes greater than 20 our predictions do not change. Hence, sample size of 20 seed entities will be used for our subsequent experiments.

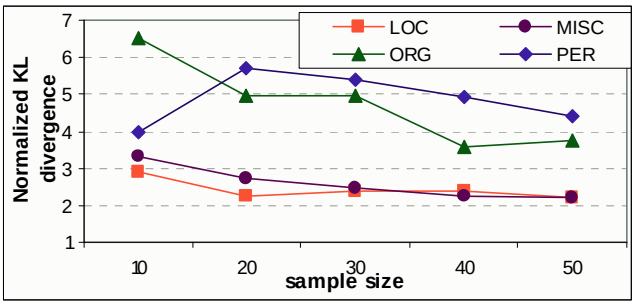

Figure 4.1: Normalized KL-divergence for context models for varying sample size (RCV 1/100, context size 2, discarding stopwords).

Figure 4.2 reports the normalized KL divergence for RCV 1/100 for language models created with N-grams of size 1, 2, and 3 words. Interestingly, the single-word (unigram) model appears to be as predictive as the two-word (bigram) and the three-word (trigram) models. In general, higher order N-gram models tend to be sparse, and hence may not be useful for our problem. Therefore, we will report results for the simpler one word models for the subsequent experiments.

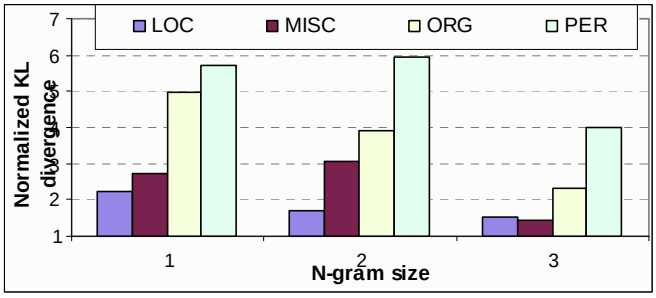

Figure 4.2: Normalized KL-divergence for context models for varying N-gram size (RCV 1/100, discarding stopwords).

We now show that our prediction technique also applies to languages other than English, by evaluating it for the named entity task on the Spanish and Dutch collections used in the CoNLL 2002 shared task evaluation [24]. As we discussed, porting information extraction systems to new domains and new languages can require significant effort. For languages other than English, annotated data and other language specific resources are less readily available. Hence, developing systems for these languages is typically more difficult, and this is also shown by the lower performance of state-of-the-art NER systems on Spanish and Dutch (Tables 4.6 and 4.7). Tables 4.6 and 4.7 report the F-measure performance of the top three systems participating in the CoNLL 2002 evaluation on the NER task for Spanish and Dutch, respectively.

	Carreras et al. [4]	Florian [14]	Cucerzan et al. [11]	Average
LOC	82.47	80.68	76.37	**79.84**
MISC	58.73	60.58	48.16	**55.82**
ORG	81.81	78.40	78.87	**79.69**
PER	88.87	86.29	85.34	**86.83**
Overall	81.39	79.05	77.15	**79.20**

Table 4.6: F-measures on the EFE newswire articles (Spanish) reported by top 3 systems participating in the CoNLL 2002 Shared Task NER competition.

	Carreras et al. [4]	Wu et al.[29]	Florian [14]	Average
LOC	79.59	80.47	77.50	**79.19**
MISC	75.41	73.04	73.25	**73.9**
ORG	71.36	67.90	69.17	**69.48**
PER	81.47	79.40	74.05	**78.31**
Overall	77.05	75.36	73.30	**75.24**

Table 4.7: F-measures on De Morgen newspaper articles (Dutch) reported by top 3 systems participating in the CoNLL 2002 Shared Task NER competition.

Tables 4.8 and 4.9 report the average KL divergence for LOC, MISC, ORG, and PER entities for the two languages by using random samples of 20 entities of each type. As we can see, our model predicts that PER entities are much easier to extract based on context than the other entities in both Spanish and Dutch.

	Context size 1		Context Size 2		Context Size 3	
LOC	2.86	**1.18**	2.53	**1.39**	2.17	**1.42**
MISC	4.19	**1.73**	3.86	**2.12**	3.60	**2.35**
ORG	3.44	**1.42**	2.90	**1.59**	2.51	**1.64**
PER	4.86	**2.01**	4.21	**2.31**	3.91	**2.56**
RANDOM	2.42		1.82		1.53	

Table 4.8: Absolute and normalized KL Divergence (averaged over 3 samples of 20 entities) for varying context sizes for the CoNLL 2002 Spanish NER task.

	Context size 1		Context Size 2		Context Size 3	
LOC	3.73	**1.44**	3.11	**1.65**	2.75	**1.61**
MISC	5.11	**1.97**	3.82	**2.02**	3.26	**1.91**
ORG	3.96	**1.53**	3.52	**1.86**	3.28	**1.92**
PER	5.82	**2.25**	4.97	**2.63**	4.44	**2.60**
RANDOM	2.59		1.89		1.71	

Table 4.9: Absolute and normalized KL Divergence (averaged over 3 samples of 20 entities) using different context sizes for the CoNLL 2002 Dutch NER task.

This is confirmed by the actual results for Spanish. For Dutch, the best performing systems in CoNLL 2002 also performed much better for PER than for ORG and MISC, LOC being, similarly to English, an outlier. Our system predicts that LOC is a difficult entity type to extract based on context for all languages mainly because many of the relevant corresponding contexts in these languages are stopwords, which occur frequently throughout the text. However, real systems were able to identify the LOC entities because the percentage of actual entities seen both in the training and in the test is typically greater than for the other entity types. This aspect of the problem is not modeled by our approach, and can be addressed in future work.

4.3 Predicting RE Difficulty

We now turn to predicting performance of relation extraction tasks (RE). The goal is to predict which relations are "difficult" to extract, and which ones are "easy". Table 4.10 reports the actual extraction accuracy on the RE task using a simple bootstrapping-based information extraction system similar to Snowball [1] and KnowItAll [12]. We report the precision on each task estimated by sampling 100 facts from the extracted relation instances. As we can see, the BORN and DIED relations are "easy" for the extraction system (exhibiting precision of as high as 97%), whereas INVENT and WROTE are relatively "hard" (exhibiting precision as low as 50%).

Relation	*Accuracy* (%) *strict*	*partial*	*Task Difficulty*
BORN	0.73	0.96	**Easy**
DIED	0.34	0.97	**Easy**
INVENT	0.35	0.64	Hard
WROTE	0.12	0.50	Hard

Table 4.10: Precision for the RE task on the Encyclopedia collection for the INVENT, BORN, DIED, and WROTE relations.

Table 4.10 reports the absolute and normalized *KL* divergence values computed from the models built by discarding common English stopwords. As we can see, the *KL* divergence values of the BORN and DIED relations are higher than the *KL* values for the INVENT and WROTE relations, predicting that the former should have higher accuracy than the latter. Hence, *KL* correctly predicts the "easy" relations vs. "hard" relations to extract. Also note that if only one word of context is considered, our model incorrectly predicts that DIED relation is more difficult than the INVENT relation. As we can see, context size of at least two words is needed to correctly predict the extraction difficulty.

Relation	Context size 1		Context size 2		Context size 3	
BORN	13.88	**2.02**	13.54	**2.17**	13.84	**2.39**
DIED	12.98	**1.89**	11.61	**1.86**	10.60	**1.83**
INVENT	13.33	**1.94**	10.92	**1.75**	9.96	**1.72**
WROTE	10.92	**1.59**	9.92	**1.59**	8.86	**1.53**
RANDOM	6.87		6.24		5.79	

Table 4.11: Absolute and normalized KL divergence for INVENT, BORN, DIED, and WROTE relations for varying context sizes (Encyclopedia, 20 sample entities, discarding stopwords).

To further investigate the required effort needed for robust prediction on the RE task, in Figure 4.3 we report the predictions for varying the seed sample size from 10 to 40 relation tuples. As we can see, our predictions remain relatively stable for sample sizes of at least 20 seed tuples. Interestingly, adding additional seed tuples beyond 20 does not improve the overall prediction accuracy as our approach: while our method distinguishes between the "easy" and "hard" relations correctly, it is not able to further distinguish between the two "hard" relations, namely that the WROTE relation is more difficult to extract than the INVENT relation.

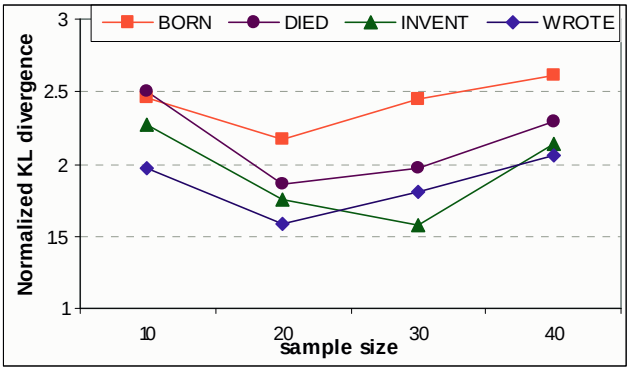

Figure 4.3: Normalized KL divergence of context models for varying sample size averaged over 3 runs for each sample size (Encyclopedia, 2-word context size, discarding stopwords).

5. CONCLUSIONS AND FUTURE WORK

We presented a general, domain and language independent approach for predicting extraction performance. We have shown that our language modeling approach is effective for predicting extraction accuracy for tasks such as named entity recognition and relation extraction, both tasks crucial for high accuracy and domain-specific information retrieval and information management.

As our experiments indicate, starting with even a small sample of available entities can be sufficient for making a reasonable

prediction about extraction accuracy. Our results are particularly encouraging as we consider a relatively simple model that does not require extra information to that typically available to modern NER and RE systems.

Extending our method to use more sophisticated language models can further improve our predictions. For languages where reliable NLP tools are available, one promising direction would be to incorporate syntactic features, and to apply techniques such as co-reference resolution to build richer and more accurate context language models. Additionally, incorporating gazetteer lists similar to those typically used by the NER systems can further improve prediction accuracy. Another interesting direction for future work is to correlate our predictions with the actual accuracy values for more fine-grained predictions.

Furthermore, our results could be applied for building interactive information extraction systems that could guide the user by requesting more examples for the extraction tasks predicted to be "difficult". Such an interactive system could more effectively focus valuable human effort on the "difficult" extraction tasks, where it is most sorely needed.

As we have shown, our approach is general and language-independent. With amounts of new information available in text increasing daily, our techniques could be extremely valuable for developing, maintaining, and deploying information extraction technology for better information access.

ACKNOWLEDGEMENTS

We thank Luis Gravano for ideas and discussions that inspired this work. We also thank Eric Brill and the anonymous referees for their insightful comments.

6. REFERENCES

[1] E. Agichtein and L. Gravano, Snowball: Extracting Relations from Large Plain-Text Collection, *in Proceedings of DL 2000*

[2] A.Bagga, Analyzing the complexity of a domain with respect to an information extraction task,*Proceedings of MUC-7*, 1998

[3] E. Brill, S. Dumais, and M. Banko. An Analysis of the AskMSR Question-Answering System, in *Proceedings of EMNLP 2002*

[4] X. Carreras, L. Márques and L. Padró, Named Entity Extraction using AdaBoost, *in Proceedings of CoNLL 2002*

[5] X. Carreras, L. Márques and L. Padró, A Simple Named Entity Extractor using AdaBoost, in *Proceedings of CoNLL 2003*

[6] C. Chelba and F. Jelinek, Exploiting Syntactic Structure for Language Modeling, in *Proceedings of COLING-ACL 1998*

[7] H. L. Chieu and H. T. Ng, Named Entity Recognition with a Maximum Entropy Approach, in *Proceedings of CoNLL 2003*

[8] M. Collins, Ranking Algorithms for Named Entity Extraction: Boosting and the Voted Perceptron, *Proceedings of ACL 2002*

[9] S. Cronen-Townsend, Y. Zhou, W. B. Croft, Predicting Query Performance, in *Proceedings of SIGIR 2002*

[10] S. Cucerzan and D. Yarowsky, Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence, in *Proceedings of EMNLP-VLC 1999*

[11] S. Cucerzan and D. Yarowsky, Language Independent NER using a Unified Model of Internal and Contextual Evidence, in *Proceedings of CoNLL 2002*

[12] O. Etzioni, M. J. Cafarella, D. Downey, S. Kok, A. M. Popescu, T. Shaked, S. Soderland, D. S. Weld and A. Yates, Web-scale information extraction in KnowItAll: preliminary results, in *Proceedings of WWW 2004*

[13] I. Dagan and O. Glickman, Probabilistic textual entailment: generic applied modeling of language variability, in *Learning Methods for Text Understanding and Mining Workshop, 2004*

[14] R. Florian, Named Entity Recognition as a House of Cards: Classifier Stacking, in *Proceedings of CoNLL 2002*

[15] R. Florian, A. Ittycheriah, H. Jing, and T. Zhang, Named Entity Recognition through Classifier Combination, in *Proceedings of CoNLL 2003*

[16] R Gaizauskas, Y Wilks, Information Extraction: Beyond Document Retrieval, *in Computational Linguistics*, 1998

[17] S. Huttunen, R. Yangarber, and R. Grishman, Complexity of event structure in IE scenarios, *Proceedings of COLING 2002*

[18] R.Jones, A.McCallum, K. Nigam, and E. Riloff, Bootstrapping for Text Learning Tasks, *IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications, 1999*

[19] D. Klein, J. Smarr, H. Nguyen, and C. D. Manning, Named Entity Recognition with Character-Level Models, in Proceedings of *CoNLL 2003*

[20] C. D. Manning and H. Schűtze, *Foundations of Statistical Natural Language Processing*, MIT Press, 1999

[21] J. Perez-Carballo and T. Strzalkowski, Natural Language Information Retrieval: Progress Report, in *Information Processing and Management Journal*,2000

[22] J. M. Ponte and W. B. Croft, A Language Modeling Approach to Information Retrieval, in *Proceedings of SIGIR 1998*

[23] H. Raghavan, J. Allan, and A. McCallum, An exploration of Entity Models, Collective Classification and Relation descriptions, in *Proceedings of LinkKDD* 2004

[24] D. Ravichandran and E. Hovy, Learning Surface Text Patterns for a Question Answering System, *Proceedings of ACL 2002*

[25] E. Riloff and R. Jones, Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping, *Proceedings of the Sixteenth National Conference on Artificial Intelligence, 1999*

[26] E. Tjong and K. Sang, Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition, *in Proceedings of CoNLL-2002*

[27] E.F. Tjong, Kim Sang and F. De Meulder, Introduction to the CoNLL-2003 Shared Task, in *Proceedings of CoNLL 2003*

[28] E.M. Voorhees, Natural Language Processing and Information Retrieval, *in Lecture Notes in Computer Science*, 1999

[29] D. Wu, G. Ngai, M. Carpuat, J. Larsen and Y. Yang, Boosting for Named Entity Recognition, *in Proceedings of CoNLL 2002*

[30] J. Xu and W. B. Croft, Improving the effectiveness of information retrieval with local context analysis, in *ACM Transactions on Information Systems, 2000*

[31] T. Zhang and D. Johnson, A Robust Risk Minimization based Named Entity Recognition System, *in Proceedings of CoNLL 2003*