Bootstrapping Language Associated with Biomedical Entities

The AID Group at TREC Genomics 2007

Edgar Meij and Sophia Katrenko

ISLA, University of Amsterdam
emeij,katrenko@science.uva.nl
http://www.adaptivedisclosure.org/

Abstract: The TREC Genomics 2007 task included recognizing topic-specific entities in the returned passages. To address this task, we have designed and implemented a novel data-driven approach by combining information extraction with language modeling techniques. Instead of using an exhaustive list of all possible instances for an entity type, we look at the language usage around each entity type and use that as a classifier to determine whether or not a piece of text discusses such an entity type. We do so by comparing it with language models of the passages. E.g., given the entity type "genes", our approach can measure the gene-iness of a piece of text.

Our algorithm works as follows. Given an entity type, it first uses Hearst patterns to extract instances of the type. To extract more instances, we look for new contextual patterns around the instances and use them as input for a bootstrapping method, in which new instances and patterns are discovered iteratively. Afterwards, all discovered instances and patterns are used to find the sentences in the collection which are most on par with the requested entity type. A language model is then generated from these sentences and, at retrieval time, we use this model to rerank retrieved passages.

As to the results of our submitted runs, we find that our baseline run performs well above the median of all participant's scores. Additionally, we find that applying our proposed method helps the entity types for which there are unambiguous patterns and numerous instances most.

1 Introduction

Our aim for this year's TREC Genomics track was to experiment with a statistical language modeling approach to entity recognition. This year's topics each contain an explicitly mentioned entity type, of which instances need to be retrieved within the returned passages. To this end, we take the results of a baseline retrieval run and rerank the passages according to the divergence of their language models with the language model of the requested entity type, which we acquire through a bootstrapping approach. Additionally, we report on a run which selects the most relevant sentences from the 10,000 highest ranking paragraphs.

The remainder of this paper is organized as follows. Section 2 describes the retrieval model we employ and the various preprocessing steps we have applied. Section 3 describes our entity recognition algorithm in more detail. In Section 4 we detail the experimental setup and we elaborate on the details of the runs we have submitted. In Section 5 we report on their performance and we end with a concluding section.

2 Retrieval Model

In all our experiments we adopt a standard query-likelihood approach [4, 6, 9]. This means that we rank documents according to their likelihood of generating the query:

$$P(d|Q) \propto P(d) \cdot \prod_{q \in Q} P(q|\theta_d), \tag{1}$$

where θ_d is a language model of document d, and q the individual query terms in query Q. The term $P(d)$ captures the prior belief in a document being relevant, which we assume to be uniform. $P(\cdot|\theta_d)$ is estimated using maximum-likelihood estimates which, in this case, means using the frequency of a query term in a document: $P(q|\theta_d) = c(q,d)/|d|$. Here, $c(q,d)$ indicates the count of term q in document d and $|d|$ the length of the particular document.

To avoid zero probabilities, we apply smoothing using a Dirichlet prior [1, 11], which is formulated as:

$$P(d|Q) \quad = \quad P(d) \cdot \prod_{q \in Q} \frac{c(q,d) + \mu P(q|\theta_C)}{|d| + \mu}, \tag{2}$$

where θ_C is the language model of a large reference corpus C (such as the collection) and μ a constant by which to tune the influence of the reference model.

3 Entity Recognition

Our working hypothesis is that we can use the language model associated with an entity type as a classifier to determine whether some piece of text discusses that entity. Instead of looking for explicit instances of a particular type, we are observing the language use around it. In other words, what is the language that people use, when they are talking about a particular entity?

In a way, this approach can be construed as a different approach to biasing relevance models. Recent work showed that biasing the generation of a query model towards query-specific MeSH terms has a positive effect on retrieval performance [7, 8]. However, instead of generating a relevance model for an entire query, we are now using a reranking approach geared specifically towards the requested entity type for a query.

The main problem is how to determine the parameters of the language model for an entity type. We approach this problem by starting out with a bootstrapping approach—which has been used succesfully for named entity recognition tasks in the past [2, 10]. In this approach, one begins with an initial pool of instances of named entities and an empty pool of contextual patterns. In each iteration, the patterns with the highest score are identified and added to the pattern pool. Further, the patterns from this pool are used to extract new entities of the same type. In our setting, we define a contextual pattern based on the immediate context of a given entity (two tokens to the right and left of it) in the documents in the collection.

We adopt a scoring scheme proposed by [10] to rank patterns and entity candidates. Given that a pattern p_i extracts W_i words, E_i of which are known entities, its score is calculated as

$$\text{score}_P(p_i) = \frac{E_i}{W_i} \cdot \log_2(E_i). \tag{3}$$

Thelen and Riloff [10] suggest adding N patterns with the highest score_P to the pattern pool. In our experiments it turned out to be sufficient to add all patterns which have a non-zero score. In addition, we also discard all patterns which consist of stop words only, since they do not provide enough evidence to be used for accurate entity recognition.

Once the patterns are added to the pattern pool, they can subsequently be used to extract new entities. An entity candidate w_i is considered to be good if it is covered by many patterns for an entity type, consequently

$$\text{score}_W(w_i) = \frac{\sum_{j=1}^{M} \log_2(E_j + 1)}{M}, \tag{4}$$

where E_j is the number of distinct entities extracted by pattern p_j and M is the number of all patterns that extract w_i. Then, the top 5 candidates are added to the entity pool. The procedure of pattern/entity selection is repeated until it reaches a certain threshold.

ENTITY TYPE	PATTERNS
GENES	*expression of * in the, of the * gene and,* *clusters of * can be*
PROTEINS	*effect on * binding to,* *cleavage of * was observed,* *associated with * and the,*
DISEASES	*episodes of * in patients,* *patients with * compared with,* *treatment of * in children*
DRUGS	*doses of * in human, effect of * therapy on,*
MUTATIONS	*if the * mutation is, that the * mutation was*
CELL OR TISSUE TYPE	*cells and * in vivo, in the * cell layer,* *studies of * maturation have*
STRAINS	*in the * strain is, bred to * mice to*
SIGNS OR SYMPTOMS	*recovery from * can take, that the * is caused*

Table 1: Examples of patterns

As there is no information as to which entities are frequent enough to start the bootstrapping process, we use Hearst patterns [3] to extract the initial list of entities. The Hearst patterns we employ are the following: *such [ENTITY TYPE]s as *, [ENTITY TYPE]s such as *, * is a [ENTITY TYPE], * and other [ENTITY TYPE]s, [ENTITY TYPE]s including *, [ENTITY TYPE]s, especially **. In the abovementioned patterns, the wildcard stands for instances of the entity. We use no shallow parsing and, as a result, multi-word entities are not considered.

Some examples of the final patterns per entity type are given in Table 1. It can be observed that some patterns are quite specific, whereas other refer to the entities of more than one topic. For instance, *that the * is caused* can be used in context of a disease name as well as in the context of the symptoms. Such ambiguous patterns might cause problems while creating a language model of a given topic. In Section 5 we provide some per-topic details as to the results of this approach.

Now that we have a set of patterns and entities per entity type $\mathcal{E}$, we retrieve the S most relevant sentences from the collection and create a language model by sampling i.i.d. from them:

$$P(t|\theta_{\mathcal{E}}) = \sum_{s \in S} P(s|\theta_{\mathcal{E}}) \cdot P(t|s), \tag{5}$$

where t denotes a vocabulary term. Then, at retrieval time, we use this model as a classifier by reranking an initial set of passages d according to the KL-divergence with this model:

$$\mathrm{D}_{kl}(\theta_{\mathcal{E}}||\theta_d) = \sum_{t} P(t|\theta_{\mathcal{E}}) \cdot \log \frac{P(t|\theta_{\mathcal{E}})}{P(t|\theta_d)}. \tag{6}$$

4 Experimental Setup

In this section we detail the specifics of our experiments as well as our submitted runs.

4.1 Preprocessing

This year's document collection is the same as in 2006. It consists of 162,259 full-text biomedical articles, which were preprocessed as follows:

1. replace HTML entities with their ISO-Latin1 counterparts,
2. remove HTML tags,
3. remove top-level tables; these only serve navigational purposes,
4. remove citations within text,
5. remove references sections,
6. lowercase terms, and,
7. stem using a Porter stemmer.

All topics are morphologically normalized as described by [5] and stemmed using a Porter stemmer.

4.2 Passage Identification

The main task for the 2007 TREC Genomics track is passage retrieval, for which we use the paragraphs in the documents. Additionally, we experiment with a more focused approach. First, 10,000 paragraps are obtained using the query-likelihood approach with Dirichlet smoothing using Eq. 2. Then, we look at the individual sentences within those paragraphs and determine their relevance—again using Eq. 2—and the most relevant ones are returned.

4.3 Runs

The three runs we have submitted have the following characteristics:

AIDrun1 baseline run, using paragraphs only, ranked according to Eq. 2. The smoothing parameter μ in Eq. 2 is set to 100 and $P(d)$ is assumed to be uniform.

AIDrun2 same as AIDrun1, but in this run we return the most relevant sentences from the top 10,000 paragraphs, as detailed in subsection 4.2.

AIDrun3 same as AIDrun1, with the top 1,000 results reranked using the algorithm described in Section 3.

	DOCUMENT	ASPECT	PASSAGE	PASSAGE2
AIDrun1	**0.241**	**0.156**	0.064	**0.069**
AIDrun2	0.195	0.088	**0.071**	0.025
AIDrun3	0.154	0.085	0.039	0.040

Table 2: The results of our submitted runs (best scores in boldface).

5 Results and Discussion

Table 2 lists the results of our submitted runs. As is clear from this table, the baseline run performs best on all accounts, except for the PASSAGE evaluation measure. The effect on this particular measure is a clear artefact of its nature, which favours shorter passages. Figure 1 gives a visual representation of the per-topic differences for AIDrun2 versus AIDrun1 in terms of PASSAGE and PASSAGE2 MAP respectively. From these graphs it's clear what the difference is between these measures on returning sentences instead of full paragraphs.

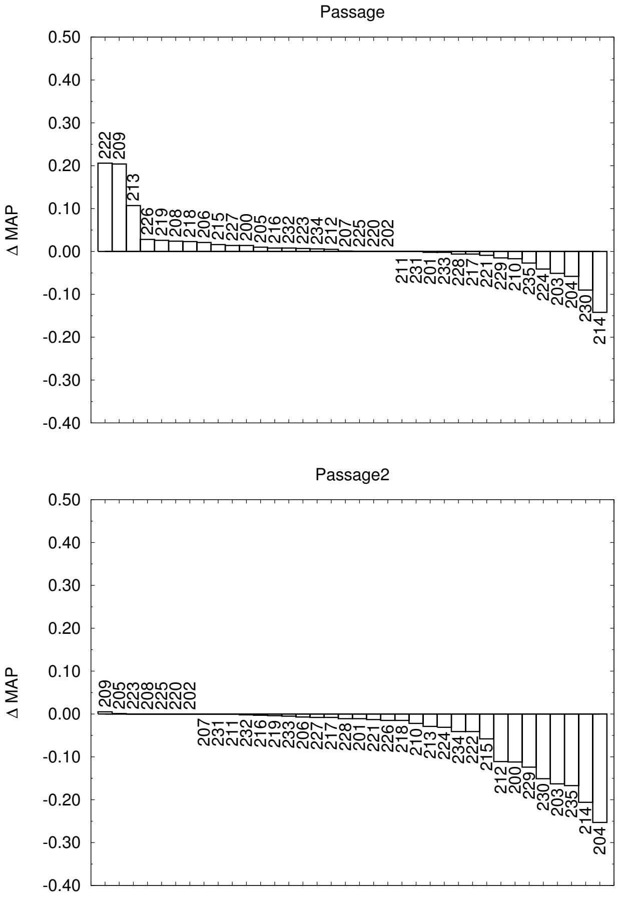

Figure 1: The difference between AIDrun2 and AIDrun1 in terms of both the passage evaluation metrics, sorted decreasingly. The labels indicate the associated topic id's.

ENTITY TYPE	DOCUMENT	ASPECT	PASSAGE	PASSAGE2
MUTATIONS	+	-	+	+
PROTEINS	±	±	±	±
GENES	±	±	±	±
DRUGS	-	-	-	-
CELL OR TISSUE TYPES	-	-	-	-
SIGNS OR SYMPTOMS	-	-	±	±
TOXICITIES	±	±	0	0
BIOLOGICAL SUBSTANCES	0	0	0	0
ANTIBODIES	-	+	0	0
DISEASES	0	0	0	0
PATHWAYS	0	0	0	0
MOLECULAR FUNCTIONS	-	-	±	±
STRAINS	-	-	0	0
TUMOR TYPES	+	-	0	0

Table 3: Impact of bootstrapping: AIDrun3 vs. AIDrun1

Figure 2 (next page) displays the difference of our baseline run, AIDrun1, as compared to the median scores of all participants. Looking at the overall picture, our run seems to improve over the median on almost all topics, except for topics 220 and 221.

Unfortunately, the retrieved instances for the entity types from the topics were not directly evaluated and, thus, we can only report on the end-to-end retrieval performance on the various measures. The results of the run employing our proposed approach to entity recognition (AIDrun3) as compared to the baseline (AIDrun1) can be found in Figure 3 (next page). Our hypothesis is that the language models for the entity types PROTEINS and GENES are the most accurate. This hypothesis is based on the results of the bootstrapping process. Protein and gene names are often mentioned in text and this results in a high number of contextual patterns. In contrast, instances of PATHWAYS or STRAINS are more difficult to detect. To verify this hypothesis, we perform a more elaborate comparison of AIDrun3 against our baseline run, AIDrun1. In Table 3, $+$ stands for the positive impact on all topics corresponding to a particular entity type, $\pm$ means a partially positive impact (on some topics but not all of them), $-$ presents a decrease on a topic, and 0 stands for no change compared against AIDrun1. As expected, PROTEINS, GENES and MUTATIONS are the topics which gain from our proposed method most. Note, however, that the distribution of queries is not uniform, i.e. some entity types are represented by one query only (ANTIBODIES, DISEASES, STRAINS and TUMOR TYPES), while some other are more frequent (e.g., PROTEINS, GENES).

6 Conclusion

For our participation in this year's TREC Genomics track, we experimented with a language modeling approach to recognizing entity types. Instead of using a more or less extensive list of possible instances, we look at the language usage associated with an entity type to detect whether or not a piece of text discusses such an entity. To this end, we have developed a model which uses a bootstrapping approach to iteratively look for new contextual patterns and instances of a particular entity type. Then, we retrieve sentences from the test collection using the found patterns and instances and construct a language model by sampling from those sentences. At retrieval time, we rerank found passages by the divergence of their respective language models with the language model of the requested entity type.

We hypothesized that our approach works best for entity types which have many unambiguous instances and contextual patterns. To test this hypothesis, we take a baseline run—which performs well above the median of all participant's scores on itself—and apply our proposed method to it. The results of this run indicate that our approach does indeed help those entity types for which there are unambiguous patterns and numerous instances most.

7 Acknowledgments

We would like to thank Maarten de Rijke, Willem van Hage, Scott Marshall, and Marco Roos for their contributions and many insightful discussions. This work was carried out in the context of the Virtual Laboratory for e-Science project (http://www.vl-e.nl). This project is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OC&W) and is part of the ICT innovation program of the Ministry of Economic Affairs (EZ).

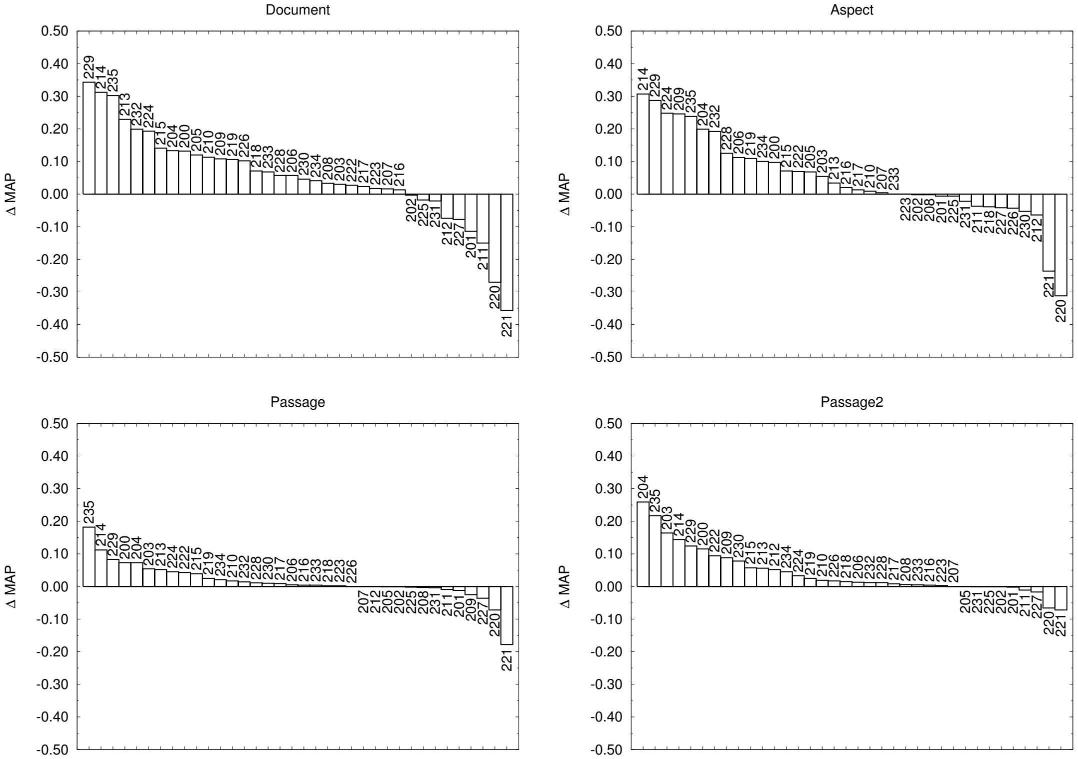

Figure 2: The difference between `AIDrun1` and the median of the scores of all participants, sorted decreasingly. The labels indicate the associated topic id's.

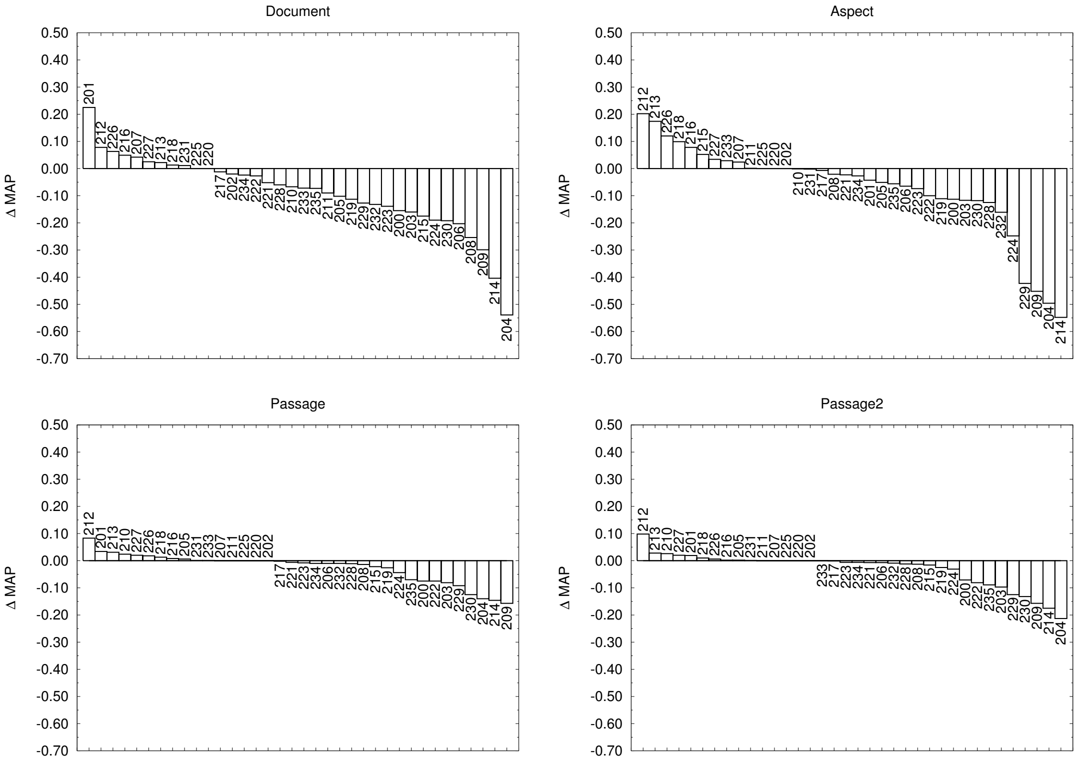

Figure 3: The difference between `AIDrun3` and `AIDrun1`, sorted decreasingly. The labels indicate the associated topic id's.

8 References

[1] Chen, S. F. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *ACL*, pages 310–318.

[2] Collins, M. and Singer, Y. (1999). Unsupervised models for named entity recognition. In *EMNLP '99*.

[3] Hearst, M. (1992). Automatic acquisition of hyponyms from large text corpora. In *COLING '92*.

[4] Hiemstra, D. (2001). *Using Language Models for Information Retrieval*. PhD thesis, University of Twente.

[5] Huang, X., Ming, Z., and Si, L. (2005). York University at TREC 2005 Genomics track. In *Proceedings of the 14th Text Retrieval Conference*.

[6] Kraaij, W. (2004). *Variations on Language Modeling for Information Retrieval*. PhD thesis, University of Twente.

[7] Meij, E. and de Rijke, M. (2007a). Integrating conceptual knowledge into relevance models: A model and estimation method. In *International Conference on the Theory of Information Retrieval (ICTIR 2007)*.

[8] Meij, E. and de Rijke, M. (2007b). Thesaurus-based feedback to support mixed search and browsing environments. In *Proceedings of the 11th European Conference on Digital Libraries (ECDL 2007)*.

[9] Ponte, J. M. and Croft, W. B. (1998). A language modeling approach to information retrieval. In *SIGIR '98*.

[10] Thelen, M. and Riloff, E. (2002). A boostrapping method for learning semantic lexicons using extraction pattern contexts. In *EMNLP '02*.

[11] Zhai, C. and Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01*.