UVA: Language Modeling Techniques for Web People Search

Krisztian Balog
ISLA, University of Amsterdam
kbalog@science.uva.nl

Leif Azzopardi
University of Glasgow
leif@dcs.gla.ac.uk

Maarten de Rijke
ISLA, University of Amsterdam
mdr@science.uva.nl

Abstract

In this paper we describe our participation in the SemEval 2007 Web People Search task. Our main aim in participating was to adapt language modeling tools for the task, and to experiment with various document representations. Our main finding is that single pass clustering, using title, snippet and body to represent documents, is the most effective setting.

1 Introduction

The goal of the Web People Search task at SemEval 2007 was to disambiguate person names in a web searching scenario (Artiles et al., 2007). Participants were presented with the following setting: given a list of documents retrieved from a web search engine using a person's name as a query, group documents that refer to the same individual.

Our aim with the participation was to adapt language modeling techniques to this task. To this end, we employed two methods: *single pass clustering* (SPC) and *probabilistic latent semantic analysis* (PLSA). Our main finding is that the former leads to high purity, while the latter leads to high inverse purity scores. Furthermore, we experimented with various document representations, based on the snippets and body text. Highest overall performance was achieved with the combination of both.

The remainder of the paper is organized as follows. In Section 2 we present the two approaches we employed for clustering documents. Next, in Section 3 we discuss document representation and preprocessing. Section 4 reports on our experiments. We conclude in Section 5.

2 Modeling

2.1 Single Pass Clustering

We employed single pass clustering (Hill., 1968) to automatically assign pages to clusters, where we assume that each cluster is a set of pages related to one particular sense of the person.

The process for assignment was performed as follows: The first document was taken and assigned to the first cluster. Then each subsequent document was compared against each cluster with a similarity measure based on the log odds ratio (initially, there was only the initial one created). A document was assigned to the most likely cluster, as long as the similarity score was higher than a threshold α; otherwise, the document was assigned to a new cluster, unless the maximum number of desired clusters η had been reached; in that case the document was assigned to the last cluster (i.e., the left overs).

The similarity measure we employed was the log odds ratio to decide whether the document was more likely to be generated from that cluster or not. This approach follows Kalt (1996)'s work on document classification using the document likelihood by representing the cluster as a multinomial term distribution (i.e., a cluster language model) and predicting the probability of a document D, given the cluster language model, i.e., $p(D|\theta_C)$. It is assumed that the terms t in a document are sampled *independently and identically*, so the log odds ratio is calculated as

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 468–471,
Prague, June 2007. ©2007 Association for Computational Linguistics

follows:

$$\log O(D,C) = \log \frac{p(D|\theta_C)}{p(D|\theta_{\bar{C}})} \quad (1)$$
$$= \log \frac{\prod_{t \in D} p(t|\theta_C)^{n(t,D)}}{\prod_{t \in D} p(t|\theta_{\bar{C}})^{n(t,D)}},$$

where $n(t, D)$ is the number of times a term appears in a document, and the $\theta_{\overline{C}}$ represents the language model that represents not being in the cluster. Note this is similar to a well-known relevance modeling approach, where the clusters are relevance and non-relevance, except, here, it is applied in the context of classification as done by Kalt (1996).

The cluster language model was estimated by performing a linear interpolation between the empirical probability of a term occurring in the cluster $p(t|C)$ and the background model $p(t)$, the probability of a term occurring at random in the collection, i.e., $p(t|\theta_C) = \lambda \cdot p(t|C) + (1 - \lambda) \cdot p(t)$, where λ was set to 0.5.[1] The "not in the cluster" language model was approximated by using the background model $p(t)$. The similarity threshold above (used for deciding whether to assign a document to an existing cluster) was set to $\alpha = 1$, and η was set to 100.

2.2 Probabilistic Latent Semantic Analysis

The second method for disambiguation we employed was probabilistic latent semantic analysis (PLSA) (Hofmann, 1999). PLSA clusters documents based on the term-document co-occurrence which results in semantic decomposition of the term document matrix into a lower dimensional latent space. Formally, PLSA can be defined as:

$$p(t, d) = p(d) \sum_z p(t|z)p(z|d), \quad (2)$$

where $p(t, d)$ is the probability of term t and document d co-occurring, $p(t|z)$ is the probability of a term given a latent topic z and $p(z|d)$ is the probability of a latent topic in a document. The prior probability of the document, $p(d)$, was assumed to be uniform. This decomposition can be obtained automatically using the EM algorithm (Hofmann, 1999). Once estimated, we assumed that each latent topic represents one of the different senses of the person, so the document is assigned to one of the person-topics. Here, we made the assignment based on the maximum $p(z|d)$, so if $p(z|d) = \max p(z|d)$, then d was assigned to z.

In order to automatically select the number of person-topics, we performed the following process to decide when the appropriate number of person-topics (defined by k) have been identified: (1) we set $k = 2$ and computed the log-likelihood of the decomposition on a held out sample of data; (2) we incremented k and computed the log-likelihood; if the log-likelihood had increased over a given threshold (0.001) then we repeated step 2, else (3) we stopped as we have maximized the log-likelihood of the decompositions, with respect to the number person-topics. This point was assumed to be the optimal with respect to the number of person senses. Since, we are focusing on identifying the true number of classes, this should result in higher inverse purity, whereas with the single pass clustering the number of clusters is not restricted, and so we would expect single pass clustering to produce more clusters but with a higher purity.

We used Lemur[2] and the PennAspect implementation of PLSA (Schein et al., 2002) for our experiments, where the parameters for PLSA where set as follows. For each k we performed 10 initializations where the best initialization in terms of log-likelihood was selected. The EM algorithm was run using tempering with up to 100 EM Steps. For tempering the setting suggested in (Hofmann, 1999) were used. The models were estimated on 90% of the data and 10% of the data was held out in order to compute the log-likelihood of the decompositions.

3 Document Representation

This section describes the various document representations we considered, and preprocessing steps we applied.

For each document, we considered the *title*, *snippet*, and *body* text. Title and snippet were provided by the output of the search engine results (`person_name.xml` files), while the body text was extracted from the crawled `index.html` files.

[1] This value was not tuned but selected based on best performing range suggested by Lavrenko and Croft (2001).

[2] `http://www.lemurproject.org`

Method	Title+Snippet				Body				Title+Snippet+Body			
	Pur	InvP	$F_{0.5}$	$F_{0.2}$	Pur	InvP	$F_{0.5}$	$F_{0.2}$	Pur	InvP	$F_{0.5}$	$F_{0.2}$
Train data												
SPC	0.903	0.298	0.422	0.336	0.776	0.416	0.482	0.434	0.768	0.438	0.506	0.456
PLSA	0.589	0.833	0.636	0.716	0.591	0.656	0.563	0.592	0.579	0.724	0.588	0.641
Test data												
SPC	0.867	0.541	0.640	0.575	0.818	0.570	0.647	0.596	0.810	0.607	0.669	0.628
PLSA	0.292	0.892	0.383	0.533	0.311	0.869	0.413	0.563	0.305	0.923	0.405	0.566

Table 1: Results of the clustering methods using various document representations.

3.1 Acquiring Plain-Text Content from HTML

Our aim is to extract the plain-text content from HTML pages and to leave out blocks or segments that contain little or no useful textual information (headers, footers, navigation menus, adverts, etc.). To this end, we exploit the fact that most web-pages consist of blocks of text content with relatively little markup, interspersed with navigation links, images with captions, etc. These segments of a page are usually separated by block-level HTML tags. Our extractor first generates a syntax tree from the HTML document. We then traverse this tree while bookkeeping the stretch of uninterrupted non-HTML text we have seen. Each time we encounter a block-level HTML tag we examine the buffer of text we have collected, and if it is longer than a threshold, we output it. The threshold for the minimal length of buffer text was empirically set to 10. In other words, we only consider segments of the page, separated by block-level HTML tags, that contain 10 or more words.

3.2 Indexing

We used a standard (English) stopword list but we did not apply stemming. A separate index was built for each person, using the Lemur toolkit. We created three index variations: `title+snippet`, `body`, and `title+snippet+body`.

In our official run we used the `title+snippet+body` index; however, in the next section we report on all three variations.

4 Results

Table 1 reports on the results of our experiments using the Single Pass Clustering (SPC) and Probabilistic Latent Semantic Analysis (PLSA) methods with various document representations. The measures (purity, inverse purity, and F-score with $\alpha = 0.5$ and $\alpha = 0.2$) are presented for both the train and test data sets.

The results clearly demonstrate the difference in the behaviors of the two clustering methods. SPC assigns people to the same cluster with high precision, as is reflected by the high purity scores. However, it is overly restrictive, and documents that belong to the same person are distributed into a number of clusters, which should be further merged. This explains the low inverse purity scores. Further experiments should be performed to evaluate to which extent this restrictive behavior could be controlled by the α parameter of the method.

In contrast with SPC, the PLSA method produces far fewer clusters per person. These clusters may cover multiple referents of a name, as is witnessed by the low purity scores. On the other hand, inverse purity scores are very high, which means referents are usually not dispersed among clusters.

As to the various document representations, we found that highest overall performance was achieved with the combination of title, snippet, and body text.

Since the data was not homogenous, it would be interesting to see how performance varies on the different names. We leave this analysis to further work.

Our official run employed the SPC method, using the `title+snippet+body` index. The results of our official submission are presented in Table 2. Our purity score was the highest of all submissions, and our system was ranked overall 4th, based on the $F_{\alpha=0.5}$ measure.

	Pur	InvP	$F_{0.5}$	$F_{0.2}$
Lowest	0.30	0.60	0.40	0.55
Highest	0.81	0.95	0.78	0.83
Average	0.54	0.82	0.60	0.69
UVA	0.81	0.60	0.67	0.62

Table 2: Official submission results and statistics.

5 Conclusions

We have described our participation in the SemEval 2007 Web People Search task. Our main aim in participating was to adapt language modeling tools for the task, and to experiment with various document representations. Our main finding is that single pass clustering, using title, snippet and body to represent documents, is the most effective setting.

We explored the two very different clustering schemes with contrasting characteristics. Looking forward, possible improvements might be pursued by combining the two approaches into a more robust system.

6 Acknowledgments

Krisztian Balog was supported by the Netherlands Organization for Scientific Research (NWO) under project number 220-80-001. Maarten de Rijke was supported by NWO under project numbers 017.001.190, 220-80-001, 264-70-050, 354-20-005, 600.065.120, 612-13-001, 612.000.106, 612.066.-302, 612.069.006, 640.001.501, 640.002.501, and by the E.U. IST programme of the 6th FP for RTD under project MultiMATCH contract IST-033104.

References

J. Artiles, J. Gonzalo, and S. Sekine. 2007. The SemEval-2007 WePS Evaluation: Establishing a benchmark for the Web People Search Task. In *Proceedings of Semeval 2007, Association for Computational Linguistics*.

D. R. Hill. 1968. A vector clustering technique. In Samuelson, editor, *Mechanised Information Storage, Retrieval and Dissemination*, North-Holland, Amsterdam.

Thomas Hofmann. 1999. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm.

T. Kalt. 1996. A new probabilistic model of text classification and retrieval. Technical Report CIIR TR98-18, University of Massachusetts, January 25, 1996.

V. Lavrenko and W. B. Croft. 2001. Relevance-based language models. In *Proceedings of the 24th annual international ACM SIGIR conference*, pages 120–127, New Orleans, LA. ACM Press.

Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. 2002. Methods and metrics for cold-start recommendations. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260, New York, NY, USA. ACM Press. See `http://www.cis.upenn.edu/datamining/software_dist/PennAspect/`.