Arabic Document Topic Analysis

Thorsten Brants, Francine Chen, Ayman Farahat

Palo Alto Research Center (PARC)
3333 Coyote Hill Rd, Palo Alto, CA 94304, USA
{brants,fchen,farahat}@parc.com

Abstract

We adopt algorithms for document topic analysis, consisting of segmentation and topic identification, to Arabic. By doing so, we outline the requirements for Arabic language resources that facilitate building, training, and fine-tuning systems that perform these tasks. Our segmentation and topic identification algorithm is based on Probabilistic Latent Semantic Analysis. First results for segmenting Arabic texts are reported.

1. Introduction

Document topic analysis is the task of assigning one or more topics to a document, characterizing the sub-topics discussed in the document, and identifying boundaries between segments discussing the different sub-topics. Most of the work in text retrieval has been on identifying and ranking the most relevant documents, although there is also work on passage retrieval. Topic analysis has applications in enabling retrieval at a finer grain than at the document level, but at a broader level than a passage.

One step in document topic analysis is topic-based segmentation. This task has been addressed by several authors. All methods calculate the similarity between the text before and after a hypothetical segment boundary and assume a segment boundary if the similarity value is small. Hearst (Hearst, 1997) describes TextTiling. She uses a sliding window and computes similarities between adjacent blocks based on their term frequency vectors. Li and Yamanishi (Li and Yamanishi, 2000a; Li and Yamanishi, 2000b) present a structured Finite Mixture Model, which they refer to as a stochastic topic model (STM). Choi et al. (Choi, 2000; Choi et al., 2001) present a model based on Latent Semantic Indexing (LSI) and divisive clustering. We have developed a segmentation method that uses the Probabilistic Latent Semantic Analysis (PLSA) model (Hofmann, 1999) for smoothing the term frequency vectors in a way that better models synonomous terms.

Topic-based segmentation is different from finding story boundaries in the TDT program. There, segmentation is not necessarily topic based, but also can (and commonly does) utilize a large variety of cue phrases which are usually absent in topic-based segmentation.

Figure 1 shows an example topic analysis for a part of an article that appeared in the El Hayat newspaper (the complete article is too long to be printed as an example). Our segmentation algorithm identified two segments, which are represented as non-underlined as underlined text. The first segment is about Israeli military operations in the West Bank, the second segment is about international efforts to defuse the tension. The next steps in topic analysis are topic identification and keyword or key phrase generation. Possible keywords are given to the right of the text (first segment: Israel, occpupy, Palestine; second segment: withdraw, stop, international).

2. Training for Arabic Document Topic Analysis

In this section, we outline the resources that are currently available for performing Arabic document topic analysis, and the resources we ideally would like to have.

2.1. Morphology

Algorithms for English document topic analysis usually depend on a morphological analyzer that associates each full-form of a word with its base form or stem. This significantly decreases the number of distinct word forms in a text by uniquely mapping a full form to some base form.

Stemming Arabic is much more difficult than stemming English. Reduction to roots can be done uniquely in the majority of cases but this would yield a very coarse grained model because words with only remotely connected meanings often share the same root. Reduction to stems (i.e., a root and a pattern) is done by the analyzer presented in (Beesley, 1996), but the output at this level is very ambiguous because of the omission of vowels in writing and the existence of diacritics and clitics. Some researchers resorted to the use of character n-grams instead of words for statistical Arabic models (Sawaf et al., 2001). Systems for uniquely identifying clitics and stems for Arabic are highly desirable as a preprocessing step for document topic analysis.

Preferable resource: Corpus of modern standard Arabic, labeld with uniquely identified stems (root and pattern) and clitics.

2.2. Segmentation

Current segmentation algorithms are trained unsupervised, i.e., no training data with explicitly labelled segment boundaries is provided. But evaluation requires such data. In the absence of documents labelled with segment boundaries, developers of segmentation algorithms use concatenated documents and try to identify the document boundaries (Choi, 2000; Hearst, 1997; Li and Yamanishi, 2000a). However, this is sub-optimal since segment boundaries *within* a document are expected to represent smaller topic shifts than boundaries *between* documents. We expect that the accuracy of a system evaluated on real document segments is lower than on artificially concatenated documents.

Preferable resource: Corpus of modern standard Arabic, labeled with segment boundaries within documents.

Figure 1: Example topic analysis for a document from the El Hayat newpaper, Apr. 11, 2002. Our segmentation algorithm TopSeg-C identified two parts. The first segment (non-underlined) is about Israeli military operations in the West bank, the second segment is about international efforts to defuse the tension. Keywords to identify the topic of the different segments are given to the right of the text.

2.3. Topic Identification

For English, collections labelled with large numbers of topics are available, e.g., in the Reuters-21578 corpus, each document is labelled with one or more of 90 different topics. Such a corpus is currently unavailable for Arabic. ELRA recently made available a collection of documents that are organized in seven domains. TREC-2001 made a step towards more detailed topics giving 25 topic descriptions but only a small number of documents (those necessary for TREC-2001) were manually labelled. Arabic topic analysis systems would benefit from large collections annotated with more fine-grained topics. This would allow topic identification and keyword evaluation as presented in (Li and Yamanishi, 2000a).

Preferable resource: Corpus of modern standard Arabic manually labeled with a fine-grained set of topic labels and keywords for each document as a whole, and for each segment in each document.

3. Topic Based Segmentation

3.1. TopSeg

TopSeg, our text segmentation system, combines the use of the Probabilistic Latent Semantic Analysis (PLSA) model (Hofmann, 1999) with the method of selecting segmentation points based on the similarity values between pairs of adjacent blocks. PLSA represents the joint probability of a document d and a word w based on a latent class variable z:

$$P(d,w) = P(d)\sum_{z} P(w|z)P(z|d) \qquad (1)$$

A model is fitted to a training corpus $\mathcal{D}$ using the Expectation-Maximization algorithm (EM) to maximize the log-likelihood function $\mathcal{L}$:

$$\mathcal{L} = \sum_{d\in\mathcal{D}}\sum_{w\in d} f(d,w)\,log P(d,w). \qquad (2)$$

After a model is trained, the model parameters $P(w|z)$ obtained in the training process are used in the process of folding-in the new (test) documents into the PLSA model to calculate $P(z|q)$ for new documents q. In the folding-in process, the Expectation-Maximization algorithm is used in a similar manner to the training process: the E-step is identical, in the M-step $P(w|z)$ is constant for all w and $P(z|q)$ is recalculated at each iteration. Usually, a very small number of iterations is sufficient for folding-in.

To segment a document, the document is first preprocessed by tokenizing the document and identifying sen-

tence boundaries. For English, two additional steps, downcasing and stemming, are performed. Next the text is broken in *blocks* of sentences. Candidate points of segmentation are identified and correspond to the locations between the text blocks. In our case, blocks are overlapping (as in a sliding window) and consist of h (e.g., $h = 5$) consecutive sentences.

Folding-in is then performed on each block b to compute the distribution among the set of latent classes, $P(z|b)$, where z is a latent variable, and b is a block. The estimated distribution of words for each block b, $P(w|b)$, is then computed as

$$P(w|b) = \sum_z P(w|z)P(z|b) \quad (3)$$

for all words w, where $P(w|z)$ is taken from the PLSA clustering of the training documents. The distribution of words in adjacent blocks b_l and b_r is compared using a similarity metric based on the Hellinger distance (Basu et al., 1997), also known as the Bhattacharyya distance (Kailath, 1967):

$$\mathrm{sim}_{\mathrm{Hel}}(b_l, b_r) = \sum_w \sqrt{P(w|b_l)P(w|b_r)}. \quad (4)$$

Dips are local minima in the similarity of adjacent blocks. We expect larger dips to correspond to stronger changes in topic. In our evaluation task, the number of segments is known is in advance, and we select the locations of the largest dips as segementation points.

3.2. TopSeg Using Combined Models

Training a PLSA model using EM starting with a random initialization yields a locally optimal model that is reached from the given start position. In general, different initializations yield different locally optimal models, which in turn might yield significantly different segmentation error rates.

One possibility to reduce the effect of different initializations is to generate several PLSA models, each with a different initializations. Then similarity values between adjacent blocks are computed according to the different models, and the resulting similarity values are averaged, yielding an *averaged similarity* curve.

The algorithm for combined models, TopSeg-C, generates k different PLSA models from the same training set, starting with different initializations. Each of the k models is used for folding-in the blocks of the test documents, and similarities between the blocks are calculated according to the k models. Now, the k similarity values for each pair of adjacent blocks are combined by calculating the average similarity value, yielding the average similarity curve. Local minima (dips) are determined in this resulting curve, and the largest dips are identified as the segment boundaries.

Similarly, we can use PLSA models with different numbers of latent classes to generate an averaged similarity curve. This was suggested by (Hofmann, 1999). However, we found that averaging over different initializations (with the same number of latent classes) yields slightly better results.

Table 1: The two corpora used in the experiments.

	Reuters-21578 (training set)	Arabic (training set)
Corpus		
# documents	7,769	6,482
# tokens	1,156,828	1,156,156
# types	41,343	70,148
# topics	90	–
Vector Space Model		
# terms	22,142	67,270
# terms $f > 1$	11,042	38,358
# n-grams	–	222,986
# n-grams $f > 1$	–	133,362

4. Topic-Based Segmentation Experiments

We performed first segmentation tests for Arabic using TextTiling and our PLSA-based model, TopSeg. Experiments and results are reported in this section. The TextTiling experiments serve as a baseline for our new segmentation model that we are currently developing for Arabic.

4.1. Data

Most of the resources outlined in section 2. are not available yet. We therefore resort to basic preprocessing and evaluation methods for performing the task of Arabic topic-based segmentation.

We prepare Arabic documents in a similar manner as Li & Yamanishi (Li and Yamanishi, 2000a) prepared documents from the Reuters-21578 corpus. 500 test documents are generated by randomly choosing two documents from the AFP Arabic Newswire Corpus (year 1994) and concatenating them into one. The task is to detect the document boundary. The system uses 6,482 documents for training[1] (training and test set are disjoint). Information about the Reuters-21578 set and the Arabic set that we prepared are provided in table 1. The sizes of the data sets are roughly comparable, but with the Arabic documents longer on average. The difference in the number of terms is even larger since we applied stemming to the English data.

Optimal values for the block size h for each model and the number of clusters Z for TopSeg and TopSeg-C were determined in preliminary experiments. For the following experiments, we set $h = 6$ for TextTiling, and $h = 5$, $Z = 256$ for TopSeg and TopSeg-C.

4.2. Results

We use the probabilistic error measure suggested by Beeferman et al. (1999) to report the results of our experiments. It is the probability p_{err}^{kw} that two *words* at distance k_w words are incorrectly identified to belong to the same/to different segments. For comparison, we present segmentation results on English data using TextTiling and STMs,

[1]The AFP Arabic Newswire Corpus is available from the Linguistic Data Consortium. The document identifiers of our concatenation of the training and test documents are available at `http://www.parc.com/istl/groups/qca/arabic-data/`

Table 2: Segmentation sentence error rate. Results marked with * are taken from (Li and Yamanishi, 2000a; Li and Yamanishi, 2000b).

Corpus	Algorithm	Terms	p_{err}^{kw}	p_{err}^{ks}
* Reuters-21578	TextTiling	stems	-	8.5 %
* Reuters-21578	STM	stems	-	9.2 %
AFP Arabic	TextTiling	fullform	8.09%	9.40%
AFP Arabic	TextTiling	n-grams	5.83%	7.49%
AFP Arabic	TopSeg	fullform	3.10%	3.88%
AFP Arabic	TopSeg	n-grams	3.05%	3.94%
AFP Arabic	TopSeg-C	fullform	**2.26%**	**2.91%**
AFP Arabic	TopSeg-C	n-grams	**2.30%**	**2.94%**

which were given in (Li and Yamanishi, 2000a; Li and Yamanishi, 2000b). They used a slightly different measure, i.e., the probability p_{err}^{ks} that two *sentences* at distance k_s sentences are incorrectly identified to belong to the same/to different segments. We therefore give both measures, p_{err}^{kw} and p_{err}^{ks}, for our results. k_w and k_s are set to be half the average length (in words and sentences, respectively) of a segment.

Table 2 presents the results on English and Arabic documents. We compare three different algorithms: TextTiling, our new algorithm using PLSA (TopSeg), and its variant using several PLSA models that are combined (TopSeg-C). Each of the three algorithms is run using full forms and using n-grams. For TextTiling, n-grams yield significantly better results than full forms (5.83% vs. 8.09% word based segmentation error rate). TopSeg yields almost identical results for n-grams and full forms. This may be explained by the property of PLSA to cluster semantically similar words, which is absent in the TextTiling algorithm. Results for using stems in Arabic are unknown yet, since no Arabic stemmer producing unique stems was available to us. In order to avoid variation that is due to different initializations of the PLSA models, we repeated each experiment using single models four times and report averaged results.

Combined models (using four different initializations) perform significantly better than single PLSA models. The word based error rates are 2.26% vs. 3.10% for full forms, and 2.30% vs. 3.05% for n-grams. Each experiment using four different random initializations is repeated four times, averaged results are reported.

Overall, TopSeg and TopSeg-C perform much better than TextTiling. The best result of 2.26% is a 61% reduction in error rate compared to TextTiling using n-grams.

Error rates for TopSeg using full forms and TopSeg using n-grams are almost identical. However, processing fullforms is much faster because the vocabulary generated from the training set only contains 38,358 different full forms, while it contains 133,362 different n-grams with $f > 1$. Computation times on a 1.7 GHz Pentium-III running Linux are as follows. For full forms, training one PLSA model with 256 classes and 20 EM iterations takes approx. 2 minutes, performing segmentation on the Arabic test set with 500 documents takes approx. 13 minutes. For n-grams, training takes approx. 9 minutes, segmentation approx. 52 minutes.

5. Conclusion

Ideally, Arabic document topic analysis would be based on a uniquely identified stem for each word, on a training collection with long documents with manually assigned segment boundaries, on manually assigned topic labels, and on manually assigned keywords word the document and its segment. Until such resorces are available, we use unlabeled documents and either full-forms or character n-grams instead. We applied our segmentation system TopSeg to Arabic newswire texts, yielding a 61% error reduction compared to TextTiling, a state-of-the-art approach for English. Our best system, using a combination of PLSA models with different random initializations, achieved an error rate of 2.26%. The system achieved approximately the same error rate when using full forms and when using n-grams as terms.

6. References

Ayanendranath Basu, Ian R. Harris, and Srabashi Basu. 1997. Minimum distance estimation: The approach using density-based distances. In G. S. Maddala and C. R. Rao, editors, *Handbook of Statistics*, volume 15, pages 21–48. North-Holland.

Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34:177.

Kenneth R. Beesley. 1996. Arabic finite-state morphological analysis and generation. In *Proceedings of COLING-96*.

Freddy Y. Y. Choi, Peter Wiemer-Hastings, and Johanna Moore. 2001. Latent semantic analysis for text segmentation. In Lillian Lee and Donna Harman, editors, *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 109–117.

Freddy Choi. 2000. Advances in domain independent linear text segmentation. In *Proceedings of NAACL-2000*, pages 26–33, Seattle, WA.

Marti A. Hearst. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of SIGIR-99*, pages 35–44, Berkeley, CA.

T. Kailath. 1967. The divergence and bhattacharyya distance measures in signal selection. *IEEE Trans. Commun. Tech.*, COM-15:52–60.

Hang Li and Kenji Yamanishi. 2000a. Topic analysis using a finite mixture model. In *Proceedings of Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 35–44.

Hang Li and Kenji Yamanishi. 2000b. Topic analysis using a finite mixture model. *IPSJ SIGNotes Natural Language (NL)*, 139(009).

Hassan Sawaf, Jorg Zaplo, and Hermann Ney. 2001. Statistical classification methods for arabic news articles. In *Proceedings of the ACL/EACL Workshop on ARABIC Language Processing: Status and Prospects*, Toulouse, France.