Segment-based Hidden Markov Models for Information Extraction

Zhenmei Gu
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2l 3G1
z2gu@uwaterloo.ca

Nick Cercone
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia, Canada B3H 1W5
nick@cs.dal.ca

Abstract

Hidden Markov models (HMMs) are powerful statistical models that have found successful applications in Information Extraction (IE). In current approaches to applying HMMs to IE, an HMM is used to model text at the document level. This modelling might cause undesired redundancy in extraction in the sense that more than one filler is identified and extracted. We propose to use HMMs to model text at the segment level, in which the extraction process consists of two steps: a segment retrieval step followed by an extraction step. In order to retrieve extraction-relevant segments from documents, we introduce a method to use HMMs to model and retrieve segments. Our experimental results show that the resulting segment HMM IE system not only achieves near zero extraction redundancy, but also has better overall extraction performance than traditional document HMM IE systems.

1 Introduction

A Hidden Markov Model (HMM) is a finite state automaton with stochastic state transitions and symbol emissions (Rabiner, 1989). The automaton models a random process that can produce a sequence of symbols by starting from some state, transferring from one state to another state with a symbol being emitted at each state, until a final state is reached. Formally, a hidden Markov model (HMM) is specified by a five-tuple (S, K, Π, A, B), where S is a set of states; K is the alphabet of observation symbols; Π is the initial state distribution; A is the probability distribution of state transitions; and B is the probability distribution of symbol emissions. When the structure of an HMM is determined, the complete model parameters can be represented as $\lambda = (A, B, \Pi)$.

HMMs are particularly useful in modelling sequential data. They have been applied in several areas within natural language processing (NLP), with one of the most successful efforts in speech recognition. HMMs have also been applied in information extraction. An early work of using HMMs for IE is (Leek, 1997) in which HMMs are trained to extract gene name-location facts from a collection of scientific abstracts. Another related work is (Bikel et al., 1997) which used HMMs as part of its modelling for the name finding problem in information extraction.

A more recent work on applying HMMs to IE is (Freitag and McCallum, 1999), in which a separate HMM is built for extracting fillers for each slot. To train an HMM for extracting fillers for a specific slot, maximum likelihood estimation is used to determine the probabilities (i.e., the initial state probabilities, the state transition probabilities, and the symbol emission probabilities) associated with each HMM from labelled texts.

One characteristic of current HMM-based IE systems is that an HMM models the entire document. Each document is viewed as a long sequence of tokens (i.e., words, punctuation marks etc.), which is the observation generated from the given HMM. The extraction is performed by finding the best state sequence for this observed long token sequence constituting the whole document, and the subsequences of tokens that pass through the target filler state are extracted as fillers. We call such approaches to applying HMMs to IE at the document level as document-based HMM IE or *document HMM IE* for brevity.

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 481–488,
Sydney, July 2006. ©2006 Association for Computational Linguistics

In addition to HMMs, there are other Markovian sequence models that have been applied to IE. Examples of these models include maximum entropy Markov models (McCallum et al., 2000), Bayesian information extraction network (Peshkin and Pfeffer, 2003), and conditional random fields (McCallum, 2003) (Peng and McCallum, 2004). In the IE systems using these models, extraction is performed by sequential tag labelling. Similar to HMM IE, each document is considered to be a single steam of tokens in these IE models as well.

In this paper, we introduce the concept of extraction redundancy, and show that current document HMM IE systems often produce undesired redundant extractions. In order to address this extraction redundancy issue, we propose a segment-based two-step extraction approach in which a segment retrieval step is imposed before the extraction step. Our experimental results show that the resulting segment-based HMM IE system not only achieves near-zero extraction redundancy but also improves the overall extraction performance.

This paper is organized as follows. In section 2, we describe our document HMM IE system in which the Simple Good-Turning (SGT) smoothing is applied for probability estimation. We also evaluate our document HMM IE system, and compare it to the related work. In Section 3, we point out the extraction redundancy issue in a document HMM IE system. The definition of the extraction redundancy is introduced for better evaluation of an IE system with possible redundant extraction. In order to address this extraction redundancy issue, we propose our segment-based HMM IE method in Section 4, in which a segment retrieval step is applied before the extraction is performed. Section 5 presents a segment retrieval algorithm by using HMMs to model and retrieve segments. We compare the performance between the segment HMM IE system and the document HMM IE system in Section 6. Finally, conclusions are made and some future work is mentioned in Section 7.

2 Document-based HMM IE with the SGT smoothing

2.1 HMM structure

We use a similar HMM structure (named as HMM_Context) as in (Freitag and McCallum, 1999) for our document HMM IE system. An example of such an HMM is shown in Figure 1, in which the number of pre-context states, post-context states, and the number of parallel filler paths are all set to 4, the default model parameter setting in our system.

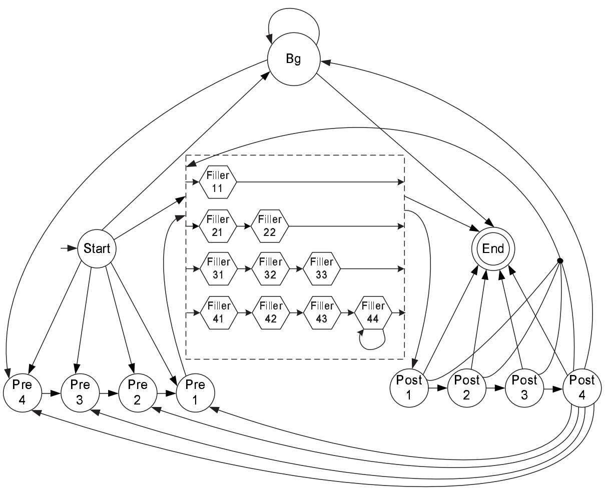

Figure 1: An example of HMM_Context structure

HMM_Context consists of the following four kinds of states in addition to the special *start* and *end* states.

Filler states $Filler_{mn}$, $m = 1, 2, 3, 4$ and $n = 1, \cdots, m$ states, correspond to the occurrences of filler tokens.

Background state This state corresponds to the occurrences of the tokens that are not related to fillers or their contexts.

Pre context states $Pre_4, Pre_3, Pre_2, Pre_1$ states correspond to the events present when context tokens occur before the fillers at the specific positions relative to the fillers, respectively.

Post context states $Post_1, Post_2, Post_3, Post_4$ states correspond to the events present when context tokens occur after the fillers at the specific positions relative to the fillers, respectively.

Our HMM structure differs from the one used in (Freitag and McCallum, 1999) in that we have added the transitions from the last post context state to every pre context state as well as every first filler state. This handles the situation where two filler occurrences in the document are so close to each other that the text segment between these two fillers is shorter than the sum of the pre context and the post context sizes.

2.2 Smoothing in HMM IE

There are many probabilities that need to be estimated to train an HMM for information extraction from a limited number of labelled documents. The data sparseness problem commonly occurring in probabilistic learning would also be an issue in the training for an HMM IE system, especially when more advanced HMM_Context models are used. Since the emission vocabulary is usually large with respect to the number of training examples, maximum likelihood estimation of emission probabilities will lead to inappropriate zero probabilities for many words in the alphabet.

The Simple Good-Turning (SGT) smoothing (Gale and Sampson, 1995) is a simple version of Good-Turning approach, which is a population frequency estimator used to adjust the observed term frequencies to estimate the real population term frequencies. The observed frequency distribution from the sample can be represented as a vector of (r, n_r) pairs, $r = 1, 2, \cdots$. r values are the observed term frequencies from the training data, and n_r refers to the number of different terms that occur with frequency r in the sample.

For each r observed in the sample, the Good-Turning method gives an estimation for its real population frequency as $r^* = (r+1)\frac{E(n_{r+1})}{E(n_r)}$, where $E(n_r)$ is the expected number of terms with frequency r. For unseen events, an amount of probability P_0 is assigned to all these unseen events, $P_0 = \frac{E(n_1)}{N} \approx \frac{n_1}{N}$, where N is the total number of term occurrences in the sample.

The SGT smoothing has been successfully applied to naive Bayes IE systems in (Gu and Cercone, 2006) for more robust probability estimation. We apply the SGT smoothing method to our HMM IE systems to alleviate the data sparseness problem in HMM training. In particular, the emission probability distribution for each state is smoothed using the SGT method. The number of unseen emission terms is estimated, as the observed alphabet size difference between the specific state emission term distribution and the all term distribution, for each state before assigning the total unseen probability obtained from the SGT smoothing among all these unseen terms.

The data sparseness problem in probability estimation for HMMs has been addressed to some extent in previous HMM based IE systems (e.g., (Leek, 1997) and (Freitag and McCallum, 1999)). Smoothing methods such as absolute discounting have been used for this purpose. Moreover, (Freitag and McCallum, 1999) uses a *shrinkage* technique for estimating word emission probabilities of HMMs in the face of sparse training data. It first defines a shrinkage topology over HMM states, then learns the mixture weights for producing interpolated emission probabilities by using a separate data set that is “held-out” from the labelled data. This technique is called *deleted interpolation* in speech recognition (Jelinek and Mercer, 1980).

2.3 Experimental results on document HMM IE and comparison to related work

We evaluated our document HMM IE system on the seminar announcements IE domain using ten-fold cross validation evaluation. The data set consists of 485 annotated seminar announcements, with the fillers for the following four slots specified for each seminar: *location* (the location of a seminar), *speaker* (the speaker of a seminar), *stime* (the starting time of a seminar) and *etime* (the ending time of a seminar). In our HMM IE experiments, the structure parameters are set to system default values, i.e., 4 for both pre-context and post-context size, and 4 for the number of parallel filler paths.

Table 1 shows F1 scores (95% confidence intervals) of our Document HMM IE system (Doc_HMM). The performance numbers from other HMM IE systems (Freitag and McCallum, 1999) are also listed in Table 1 for comparison, where HMM_None is their HMM IE system that uses absolute discounting but with no shrinkage, and HMM_Global is the representative version of their HMM IE system with shrinkage.

By using the same structure parameters (i.e., the same context size) as in (Freitag and McCallum, 1999), our Doc_HMM system performs consistently better on all slots than their HMM IE system using absolute discounting. Even compared to their much more complex version of HMM IE with shrinkage, our system has achieved comparable results on *location*, *speaker* and *stime*, but obtained significantly better performance on the *etime* slot. It is noted that our smoothing method is much simpler to apply, and does not require any extra effort such as specifying shrinkage topology or any extra labelled data for a held-out set.

Table 1: F1 of Document HMM IE systems on seminar announcements

Learner	**location**	**speaker**	**stime**	**etime**
Doc_HMM	0.8220±0.022	0.7135±0.025	1.0000±0.0	0.9488±0.012
HMM_None	0.735	0.513	0.991	0.814
HMM_Global	0.839	0.711	0.991	0.595

3 Document extraction redundancy in HMM IE

3.1 Issue with document-based HMM IE

In existing HMM based IE systems, an HMM is used to model the entire document as one long observation sequence emitted from the HMM. The extracted fillers are identified by any part of the sequence in which tokens in it are labelled as one of the filler states. The commonly used structure of the hidden Markov models in IE allows multiple passes through the paths of the filler states. So it is possible for the labelled state sequences to present multiple filler extractions.

It is not known from the performance reports from previous works (e.g., (Freitag and McCallum, 1999)) that how exactly a correct extraction for one document is defined in HMM IE evaluation. One way to define a correct extraction for a document is to require that at least one of the text segments that pass the filler states is the same as a labelled filler. Alternatively, we can define the correctness by requiring that all the text segments that pass the filler states are same as the labelled fillers. In this case, it is actually required an exact match between the HMM state sequence determined by the system and the originally labelled one for that document. Very likely, the former correctness criterion was used in evaluating these document-based HMM IE systems. We used the same criterion for evaluating our document HMM IE systems in Section 2.

Although it might be reasonable to define that a document is correctly extracted if any one of the identified fillers from the state sequence labelled by the system is a correct filler, certain issues exist when a document HMM IE system returns multiple extractions for the same slot for one document. For example, it is possible that some of the fillers found by the system are not correct extractions. In this situation, such document-wise extraction evaluation alone would not be sufficient to measure the performance of an HMM IE system.

Document HMM IE modelling does provide any guidelines for selecting one mostly likely filler from the ones identified by the state sequence matching over the whole document. For the template filling IE problem that is of our interest in this paper, the ideal extraction result is one slot filler per document. Otherwise, some further post-processing would be required to choose only one extraction, from the multiple fillers possibly extracted by a document HMM IE system, for filling in the slot template for that document.

3.2 Concept of document extraction redundancy in HMM IE

In order to make a more complete extraction performance evaluation in an HMM-based IE system, we introduce another performance measure, *document extraction redundancy* as defined in Definition 1, to be used with the document-wise extraction correctness measure .

Definition 1. Document extraction redundancy *is defined over the documents that contain correct extraction(s), as the ratio of the* **incorrectly** *extracted fillers to all returned fillers from the document HMM IE system.*

For example, when the document HMM IE system issues more than one slot extraction for a document, if all the issued extractions are correct ones, then the extraction redundancy for that document is 0. Among all the issued extractions, the larger of the number of incorrect extractions is, the closer the extraction redundancy for that document is to 1. However, the extraction redundancy can never be 1 according to our definition, since this measure is only defined over the documents that contain at lease one correct extraction.

Now let us have a look at the extraction redundancy in the document HMM IE system from Section 2. We calculate the average document extraction redundancy over all the documents that are judged as correctly extracted. The evaluation results for the document extraction redundancy (shown in column R) are listed in Table 2, paired with their corresponding F1 scores from the document-wise extraction evaluation.

Table 2: F1 / redundancy in document HMM IE on SA domain

Slot	F1	R
location	0.8220	0.0543
speaker	0.7135	0.0952
stime	1.0000	0.1312
etime	0.9488	0.0630

Generally speaking, the HMM IE systems based on document modelling has exhibited a certain extraction redundancy for any slot in this IE domain, and in some cases such as for *speaker* and *stime*, the average extraction redundancy is by all means not negligible.

4 Segment-based HMM IE Modelling

In order to make the IE system capable of producing the ideal extraction result that issues only one slot filler for each document, we propose a segment-based HMM IE framework in the following sections of this paper. We expect this framework can dramatically reduce the document extraction redundancy and make the resulting IE system output extraction results to the template filling IE task with the least post-processing requirement.

The basic idea of our approach is to use HMMs to extract fillers from only *extraction-relevant* part of text instead of the entire document. We refer to this modelling as segment-based HMM IE, or *segment HMM IE* for brevity. The unit of the extraction-relevant text segments is definable according to the nature of the texts. For most texts, one sentence in the text can be regarded as a text segment. For some texts that are not written in a grammatical style and sentence boundaries are hard to identify, we can define a *extraction-relevant* text segment be the part of text that includes a filler occurrence and its contexts.

4.1 Segment-based HMM IE modelling: the procedure

By imposing an extraction-relevant text segment retrieval in the segment HMM IE modelling, we perform an extraction on a document by completing the following two successive sub-tasks.

Step 1: Identify from the entire documents the text segments that are relevant to a specific slot extraction. In other words, the document is filtered by locating text segments that might contain a filler.

Step 2: Extraction is performed by applying the segment HMM only on the extraction-relevant text segments that are obtained from the first step. Each retrieved segment is labelled with the most probable state sequence by the HMM, and all these segments are sorted according to their normalized likelihoods of their best state sequences. The filler(s) identified by the segment having the largest likelihood is/are returned as the extraction result.

4.2 Extraction from relevant segments

Since it is usual that more than one segment have been retrieved at Step 1, these segments need to compete at step 2 for issuing extraction(s) from their best state sequences found with regard to the HMM λ used for extraction. For each segment s with token length of n, its normalized best state sequence likelihood is defined as follows.

$$l(s) = \log\bigl(\max_{all\ Q} P(Q, s|\lambda)\bigr) \times \frac{1}{n}, \qquad (1)$$

where λ is the HMM and Q is any possible state sequence associated with s. All the retrieved segments are then ranked according to their $l(s)$, and the segment with the highest $l(s)$ number is selected and the extraction is identified from its labelled state sequence by the segment HMM.

This proposed two-step HMM based extraction procedure requires that the training of the IE models follows the same style. First, we need to learn an extraction-relevance segment retrieval system from the labelled texts which will be described in detail in Section 5. Then, an HMM is trained for each slot extraction by only using the extraction-relevant text segments instead of the whole documents.

By limiting the HMM training to a much smaller part of the texts, basically including the fillers and their surrounding contexts, the alphabet size of all emission symbols associated with the HMM would be significantly reduced. Compared to the common document-based HMM IE modelling, our proposed segment-based HMM IE modelling would also ease the HMM training difficulty caused by the data sparseness problem since we are working on a smaller alphabet.

5 Extraction-relevant segment retrieval using HMMs

We propose a segment retrieval approach for performing the first subtask by also using HMMs. In particular, it trains an HMM from labelled segments in texts, and then use the learned HMM to determine whether a segment is relevant or not with regard to a specific extraction task. In order to distinguish the HMM used for segment retrieval in the first step from the HMM used for the extraction in the second step, we call the former one as the *retrieval HMM* and the later one as the *extractor HMM*.

5.1 Training HMMs for segment retrieval

To train a retrieval HMM, it requires each training segment to be labelled in the same way as in the annotated training document. After the training texts are segmented into sentences (we are using sentence as the segment unit), the obtained segments that carry the original slot filler tags are used directly as the training examples for the retrieval HMM.

An HMM with the same IE specific structure is trained from the prepared training segments in exactly the same way as we train an HMM in the document HMM IE system from a set of training documents. The difference is that much shorter labelled observation sequences are used.

5.2 Segment retrieval using HMMs

After a retrieval HMM is trained from the labelled segments, we use this HMM to determine whether an unseen segment is relevant or not to a specific extraction task. This is done by estimating, from the HMM, how likely the associated state sequence of the given segment passes the target filler states. The HMM λ trained from labelled segments has the structure as shown in Figure 1. So for a segment s, all the possible state sequences can be categorized into two kinds: the state sequences passing through one of the target filler path, and the state sequences not passing through any target filler states.

Because of the structure constraints of the specified HMM in IE, we can see that the second kind of state sequences actually have only one possible path, denoted as Q_{bg} in which the whole observation sequence of s starts at the background state q_{bg} and continues staying in the background state until the end. Let $s = O_1 O_2 \cdots O_T$, where T is the length of s in tokens. The probability of s following this particular background state path Q_{bg} can be easily calculated with respect to the HMM λ as follows:

$$\begin{aligned} P(s, Q_{bg}|\lambda) = & \pi_{q_{bg}} b_{q_{bg}}(O_1) a_{q_{bg}q_{bg}} b_{q_{bg}}(O_2) \\ & \cdots a_{q_{bg}q_{bg}} b_{q_{bg}}(O_T), \end{aligned}$$

where π_i is the initial state probability for state i, $b_i(O_t)$ is the emission probability of symbol O_t at state i, and a_{ij} is the state transition probability from state i to state j.

We know that the probability of observing s given the HMM λ actually sums over the probabilities of observing s on all the possible state sequences given the HMM, i.e.,

$$P(s|\lambda) = \sum_{all\ Q} P(s, Q|\lambda)$$

Let Q_{filler} denote the set of state sequences that pass through any filler states. We have $\{all\ Q\} = Q_{bg} \cup Q_{filler}$. $P(s|\lambda)$ can be calculated efficiently using the forward-backward procedure which makes the estimate for the total probability of all state paths that go through filler states straightforward to be:

$$\begin{aligned} P(s, Q_{filler}|\lambda) \triangleq & \sum_{allQ \in Q_{filler}} P(s, Q|\lambda) \\ = & P(s|\lambda) - P(s, Q_{bg}|\lambda). \end{aligned}$$

Now it is clear to see that, if the calculated $P(s, Q_{filler}|\lambda) > P(s, Q_{bg}|\lambda)$, then segment s is considered more likely to have filler occurrence(s). Therefore in this case we classify s as an extraction relevant segment and it will be retrieved.

5.3 Document-wise retrieval performance

Since the purpose of our segment retrieval is to identify relevant segments from each document, we need to define how to determine whether a document is correctly filtered (i.e., with extraction relevant segments retrieved) by a given segment retrieval system. We consider two criteria, first a loose correctness definition as follows:

Definition 2. *A document is* ***least correctly filtered*** *by the segment retrieval system when* ***at least one*** *of the extraction relevant segments in that document has been retrieved by the system; otherwise, we say the system fails on that document.*

Then we define a stricter correctness measure as follows:

Definition 3. *A document is* ***most correctly filtered*** *by the segment retrieval system only when* ***all*** *the extraction relevant segments in that document have been retrieved by the system; otherwise, we say the system fails on that document.*

The overall segment retrieval performance is measured by *retrieval precision* (i.e., ratio of the number of correctly filtered documents to the number of documents from which the system has retrieved at least one segments) and *retrieval recall* (i.e., ratio of the number of correctly filtered documents to the number of documents that contain relevant segments). According to the just defined two correctness measures, the overall retrieval performance for the all testing documents can be evaluated under both the *least correctly filtered* and the *least correctly filtered* measures.

We also evaluate average *document-wise segment retrieval redundancy*, as defined in Definition 4 to measure the segment retrieval accuracy.

Definition 4. ***Document-wise segment retrieval redundancy*** *is defined over the documents which are least correctly filtered by the segment retrieval system, as the ratio of the retrieved* ***irrelevant*** *segments to all retrieved segments for that document.*

5.4 Experimental results on segment retrieval

Table 3 shows the document-wise segment retrieval performance evaluation results under both *least correctly filtered* and *most correctly filtered* measures, as well as the related average number of retrieved segments for each document (as in Column **nSeg**) and the average retrieval redundancy.

Shown from Table 3, the segment retrieval results have achieved high recall especially with the *least correctly filtered* correctness criterion. In addition, the system has produced the retrieval results with relatively small redundancy which means most of the segments that are fed to the segment HMM extractor from the retrieval step are actually extraction-related segments.

6 Segment vs. document HMM IE

We conducted experiments to evaluate our segment-based HMM IE model, using the proposed segment retrieval approach, and comparing their final extraction performance to the document-based HMM IE model. Table 4 shows the overall performance comparison between the document HMM IE system (Doc_HMM) and the segment HMM IE system (Seg_HMM).

Compared to the document-based HMM IE modelling, the extraction performance on *location* is significantly improved by our segment HMM IE system. The important improvement from the segment HMM IE system that it has achieved zero extraction redundancy for all the slots in this experiment.

7 Conclusions and future work

In current HMM based IE systems, an HMM is used to model at the document level which causes certain redundancy in the extraction. We propose a segment-based HMM IE modelling method in order to achieve near-zero redundancy extraction. In our segment HMM IE approach, a segment retrieval step is first applied so that the HMM extractor identifies fillers from a smaller set of extraction-relevant segments. The resulting segment HMM IE system using the segment retrieval method has not only achieved nearly zero extraction redundancy, but also improved the overall extraction performance. The effect of the segment-based HMM extraction goes beyond applying a post-processing step to the document-based HMM extraction, since the latter can only reduce the redundancy but not improve the F1 scores.

For the template-filling style IE problems, it is more reasonable to perform extraction by HMM state labelling on segments, instead of on the entire document. When the observation sequence to be labelled becomes longer, finding the best single state sequence for it would become a more difficult task. Since the effect of changing a small part in a very long state sequence would not be as obvious, with regard to the state path probability calculation, as changing the same subsequence in a much shorter state sequence. In fact, this perspective not only applies in HMM IE modelling, but also applies in any IE modelling in which extraction is performed by sequential state labelling. We are working on extending this segment-based framework to other Markovian sequence models used for IE.

Segment retrieval for extraction is an important step in segment HMM IE, since it filters out irrelevant segments from the document. The HMM for extraction is supposed to model extraction-relevant segments, so the irrelevant segments that are fed to the second step would make the extraction more difficult by adding noise to the competition among relevant segments. We have

Table 3: Segment retrieval results

Slot	*least correctly*		*most correctly*			
	Precision	**Recall**	**Precision**	**Recall**	**nSeg**	**Redundancy**
location	0.8948	0.9177	0.8758	0.8982	2.6064	0.4569
speaker	0.8791	0.7633	0.6969	0.6042	1.6082	0.1664
stime	1.0000	1.0000	0.9464	0.9464	2.6576	0.1961
etime	0.4717	0.9952	0.4570	0.9609	1.7896	0.1050

Table 4: F1 comparison on seminar announcements (document HMM IE vs. segment HMM IE)

Learner	**location**		**speaker**		**stime**		**etime**	
	F1	R	F1	R	F1	R	F1	R
Doc_HMM	0.822±0.022	0.0543	0.7135±0.025	0.0952	1.0000±0.0	0.131	0.9488±0.012	0.063
Seg_HMM	0.8798±0.018	0	0.7162±0.025	0	0.998±0.003	0	0.9611±0.011	0

presented and evaluated our segment retrieval method. Document-wise retrieval performance can give us more insights on the goodness of a particular segment retrieval method for our purpose: the document-wise retrieval recall using the *least correctly filtered* measure provides an upper bound on the final extraction performance.

Our current segment retrieval method requires the training documents to be segmented in advance. Although sentence segmentation is a relatively easy task in NLP, some segmentation errors are still unavoidable especially for ungrammatical online texts. For example, an improper segmentation could set a segment boundary in the middle of a filler, which would definitely affect the final extraction performance of the segment HMM IE system. In the future, we intend to design segment retrieval methods that do not require documents to be segmented before retrieval, hence avoiding the possibility of early-stage errors introduced from the text segmentation step. A very promising idea is to adapt a naive Bayes IE to perform redundant extractions directly on an entire document to retrieve filler-containing text segments for a segment HMM IE system.

References

[Bikel et al.1997] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel. 1997. Nymble: a high-performance learning name-finder. In *Proceedings of ANLP-97*, pages 194–201.

[Freitag and McCallum1999] D. Freitag and A. McCallum. 1999. Information extraction with HMMs and shrinkage. In *Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction*.

[Gale and Sampson1995] W. Gale and G. Sampson. 1995. Good-turning smoothing without tears. *Journal of Quantitative Linguistics*, 2:217–37.

[Gu and Cercone2006] Z. Gu and N. Cercone. 2006. Naive bayes modeling with proper smoothing for information extraction. In *Proceedings of the 2006 IEEE International Conference on Fuzzy Systems*.

[Jelinek and Mercer1980] F. Jelinek and R. L. Mercer. 1980. Intepolated estimation of markov source parameters from sparse data. In E. S. Gelesma and L. N. Kanal, editors, *Proceedings of the Wrokshop on Pattern Recognition in Practice*, pages 381–397, Amsterdam, The Netherlands: North-Holland, May.

[Leek1997] T. R. Leek. 1997. Information extraction using hidden markov models. Master's thesis, UC San Diego.

[McCallum et al.2000] A. McCallum, D. Freitag, and F. Pereira. 2000. Maximum entropy Markov models for informaion extraction and segmentation. In *Proceedings of ICML-2000*.

[McCallum2003] Andrew McCallum. 2003. Efficiently inducing features of conditional random fields. In *Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI03)*.

[Peng and McCallum2004] F. Peng and A. McCallum. 2004. Accurate information extraction from research papers using conditional random fields. In *Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics*.

[Peshkin and Pfeffer2003] L. Peshkin and A. Pfeffer. 2003. Bayesian information extraction network. In *Proceedings of the Eighteenth International Joint Conf. on Artificial Intelligence*.

[Rabiner1989] L. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77(2).