Do Automatic Annotation Techniques Have Any Impact on Supervised Complex Question Answering?

Yllias Chali
University of Lethbridge
Lethbridge, AB, Canada
chali@cs.uleth.ca

Sadid A. Hasan
University of Lethbridge
Lethbridge, AB, Canada
hasan@cs.uleth.ca

Shafiq R. Joty
University of British Columbia
Vancouver, BC, Canada
rjoty@cs.ubc.ca

Abstract

In this paper, we analyze the impact of different automatic annotation methods on the performance of supervised approaches to the complex question answering problem (defined in the DUC-2007 main task). Huge amount of annotated or labeled data is a prerequisite for supervised training. The task of labeling can be accomplished either by humans or by computer programs. When humans are employed, the whole process becomes time consuming and expensive. So, in order to produce a large set of labeled data we prefer the automatic annotation strategy. We apply five different automatic annotation techniques to produce labeled data using ROUGE similarity measure, Basic Element (BE) overlap, syntactic similarity measure, semantic similarity measure, and Extended String Subsequence Kernel (ESSK). The representative supervised methods we use are Support Vector Machines (SVM), Conditional Random Fields (CRF), Hidden Markov Models (HMM), and Maximum Entropy (MaxEnt). Evaluation results are presented to show the impact.

1 Introduction

In this paper, we consider the complex question answering problem defined in the DUC-2007 main task[1]. We focus on an extractive approach of summarization to answer complex questions where a subset of the sentences in the original documents are chosen. For supervised learning methods, huge amount of annotated or labeled data sets are obviously required as a precondition. The decision as to whether a sentence is important enough to be annotated can be taken either by humans or by computer programs. When humans are employed in the process, producing such a large labeled corpora becomes time consuming and expensive. There comes the necessity of using automatic methods to align sentences with the intention to build extracts from abstracts. In this paper, we use ROUGE similarity measure, Basic Element (BE) overlap, syntactic similarity measure, semantic similarity measure, and Extended String Subsequence Kernel (ESSK) to automatically label the corpora of sentences (DUC-2006 data) into extract summary or non-summary categories in correspondence with the document abstracts. We feed these 5 types of labeled data into the learners of each of the supervised approaches: SVM, CRF, HMM, and MaxEnt. Then we extensively investigate the performance of the classifiers to label unseen sentences (from 25 topics of DUC-2007 data set) as summary or non-summary sentence. The experimental results clearly show the impact of different automatic annotation methods on the performance of the candidate supervised techniques.

2 Automatic Annotation Schemes

Using ROUGE Similarity Measures ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is an automatic tool to determine the quality of a summary using a collection of measures ROUGE-N (N=1,2,3,4), ROUGE-L, ROUGE-W and ROUGE-S which count the number of overlapping units such as n-gram, word-sequences, and word-pairs between the extract and the abstract summaries (Lin, 2004). We assume each individual document sentence as the extract summary and calculate its ROUGE similarity scores with the corresponding abstract summaries. Thus an average ROUGE score is assigned to each sentence in the document. We choose the top N sentences based on ROUGE scores to have the label

[1] http://www-nlpir.nist.gov/projects/duc/duc2007/

$+1$ (summary sentences) and the rest to have the label -1 (non-summary sentences).

Basic Element (BE) Overlap Measure We extract BEs, the "head-modifier-relation" triples for the sentences in the document collection using BE package 1.0 distributed by ISI [2]. The ranked list of BEs sorted according to their Likelihood Ratio (LR) scores contains important BEs at the top which may or may not be relevant to the abstract summary sentences. We filter those BEs by checking possible matches with an *abstract sentence word* or a *related word*. For each abstract sentence, we assign a score to every document sentence as the sum of its filtered BE scores divided by the number of BEs in the sentence. Thus, every abstract sentence contributes to the BE score of each document sentence and we select the top N sentences based on average BE scores to have the label $+1$ and the rest to have the label -1.

Syntactic Similarity Measure In order to calculate the syntactic similarity between the abstract sentence and the document sentence, we first parse the corresponding sentences into syntactic trees using Charniak parser [3] (Charniak, 1999) and then we calculate the similarity between the two trees using the *tree kernel* (Collins and Duffy, 2001). We convert each parenthesis representation generated by Charniak parser to its corresponding tree and give the trees as input to the tree kernel functions for measuring the syntactic similarity. The tree kernel of two syntactic trees T_1 and T_2 is actually the inner product of the two m-dimensional vectors, $v(T_1)$ and $v(T_2)$:

$$TK(T_1,T_2)=v(T_1).v(T_2)$$

The *TK* (tree kernel) function gives the similarity score between the abstract sentence and the document sentence based on the syntactic structure. Each abstract sentence contributes a score to the document sentences and the top N sentences are selected to be annotated as $+1$ and the rest as -1 based on the average of similarity scores.

Semantic Similarity Measure Shallow semantic representations, bearing a more compact information, can prevent the sparseness of deep structural approaches and the weakness of BOW models (Moschitti et al., 2007). To experiment with semantic structures, we parse the corresponding sentences semantically using a Semantic Role Labeling (SRL) system like ASSERT[4]. ASSERT is an automatic statistical semantic role tagger, that can annotate naturally occuring text with semantic arguments. We represent the annotated sentences using tree structures called semantic trees (ST). Thus, by calculating the similarity between STs, each document sentence gets a semantic similarity score corresponding to each abstract sentence and then the top N sentences are selected to be labeled as $+1$ and the rest as -1 on the basis of average similarity scores.

Extended String Subsequence Kernel (ESSK) Formally, ESSK is defined as follows (Hirao et al., 2004):

$$K_{essk}(T,U)=\sum_{m=1}^{d}\sum_{t_i\in T}\sum_{u_j\in U}K_m(t_i,u_j)$$

$$K_m(t_i,u_j)=\begin{cases} val(t_i,u_j) & \text{if } m=1 \\ K'_{m-1}(t_i,u_j)\cdot val(t_i,u_j) & \end{cases}$$

Here, $K'_m(t_i,u_j)$ is defined below. t_i and u_j are the nodes of T and U, respectively. Each node includes a word and its disambiguated sense. The function $val(t,u)$ returns the number of attributes common to the given nodes t and u.

$$K'_m(t_i,u_j)=\begin{cases} 0 & \text{if } j=1 \\ \lambda K'_m(t_i,u_{j-1})+K''_m(t_i,u_{j-1}) & \end{cases}$$

Here λ is the decay parameter for the number of skipped words. We choose $\lambda=0.5$ for this research. $K''_m(t_i,u_j)$ is defined as:

$$K''_m(t_i,u_j)=\begin{cases} 0 & \text{if } i=1 \\ \lambda K''_m(t_{i-1},u_j)+K_m(t_{i-1},u_j) & \end{cases}$$

Finally, the similarity measure is defined after normalization as below:

$$sim_{essk}(T,U)=\frac{K_{essk}(T,U)}{\sqrt{K_{essk}(T,T)K_{essk}(U,U)}}$$

Indeed, this is the similarity score we assign to each document sentence for each abstract sentence and in the end, top N sentences are selected to be annotated as $+1$ and the rest as -1 based on average similarity scores.

3 Experiments

Task Description The problem definition at DUC-2007 was: *"Given a complex question (topic description) and a collection of relevant documents, the task is to synthesize a fluent, well-organized 250-word summary of the documents*

[2] BE website:http://www.isi.edu/ cyl/BE

[3] available at ftp://ftp.cs.brown.edu/pub/nlparser/

[4] available at http://cemantix.org/assert

that answers the question(s) in the topic". We consider this task and use the five automatic annotation methods to label each sentence of the 50 document sets of DUC-2006 to produce five different versions of training data for feeding the SVM, HMM, CRF and MaxEnt learners. We choose the top 30% sentences (based on the scores assigned by an annotation scheme) of a document set to have the label $+1$ and the rest to have -1. Unlabeled sentences of 25 document sets of DUC-2007 data are used for the testing purpose.

Feature Space We represent each of the document-sentences as a vector of feature-values. We extract several query-related features and some other important features from each sentence. We use the features: n-gram overlap, Longest Common Subsequence (LCS), Weighted LCS (WLCS), skip-bigram, exact word overlap, synonym overlap, hypernym/hyponym overlap, gloss overlap, Basic Element (BE) overlap, syntactic tree similarity measure, position of sentences, length of sentences, Named Entity (NE), cue word match, and title match (Edmundson, 1969).

Supervised Systems For SVM we use second order polynomial kernel for the ROUGE and ESSK labeled training. For the BE, syntactic, and semantic labeled training third order polynomial kernel is used. The use of kernel is based on the accuracy we achieved during training. We apply 3-fold cross validation with randomized local-grid search for estimating the value of the trade-off parameter C. We try the value of C in 2^i following heuristics, where $i \in \{-5, -4, \cdots, 4, 5\}$ and set C as the best performed value 0.125 for second order polynomial kernel and default value is used for third order kernel. We use SVM^{light} [5] package for training and testing in this research. In case of HMM, we apply the Maximum Likelihood Estimation (MLE) technique by frequency counts with add-one smoothing to estimate the three HMM parameters: initial state probabilities, transition probabilities and emission probabilities. We use Dr. Dekang Lin's HMM package[6] to generate the most probable label sequence given the model parameters and the observation sequence (unlabeled DUC-2007 test data). We use MALLET-0.4 NLP toolkit[7] to implement the CRF. We formulate our problem in terms of MALLET's SimpleTagger class which is a command line interface to the MALLET CRF class. We modify the SimpleTagger class in order to include the provision for producing corresponding posterior probabilities of the predicted labels which are used later for ranking sentences. We build the MaxEnt system using Dr. Dekang Lin's MaxEnt package[8]. To define the exponential prior of the λ values in MaxEnt models, an extra parameter α is used in the package during training. We keep the value of α as default.

Sentence Selection The proportion of important sentences in the training data will differ from the one in the test data. A simple strategy is to rank the sentences in a document, then select the top N sentences. In SVM systems, we use the normalized distance from the hyperplane to each sample to rank the sentences. Then, we choose N sentences until the summary length (250 words for DUC-2007) is reached. For HMM systems, we use Maximal Marginal Relevance (MMR) based method to rank the sentences (Carbonell et al., 1997). In CRF systems, we generate posterior probabilities corresponding to each predicted label in the label sequence to measure the confidence of each sentence for summary inclusion. Similarly for MaxEnt, the corresponding probability values of the predicted labels are used to rank the sentences.

Evaluation Results The multiple "reference summaries" given by DUC-2007 are used in the evaluation of our summary content. We evaluate the system generated summaries using the automatic evaluation toolkit ROUGE (Lin, 2004). We report the three widely adopted important ROUGE metrics in the results: ROUGE-1 (unigram), ROUGE-2 (bigram) and ROUGE-SU (skip bi-gram). Figure 1 shows the ROUGE F-measures for SVM, HMM, CRF and MaxEnt systems. The X-axis containing **ROUGE**, **BE**, **Synt** (Syntactic), **Sem** (Semantic), and **ESSK** stands for the annotation scheme used. The Y-axis shows the ROUGE-1 scores at the top, ROUGE-2 scores at the bottom and ROUGE-SU scores in the middle. The supervised systems are distinguished by the line style used in the figure.

From the figure, we can see that the *ESSK* labeled SVM system is having the poorest ROUGE - 1 score whereas the *Sem* labeled system performs

[5] http://svmlight.joachims.org/

[6] http://www.cs.ualberta.ca/ ̃lindek/hmm.htm

[7] http://mallet.cs.umass.edu/

[8] http://www.cs.ualberta.ca/ ̃lindek/downloads.htm

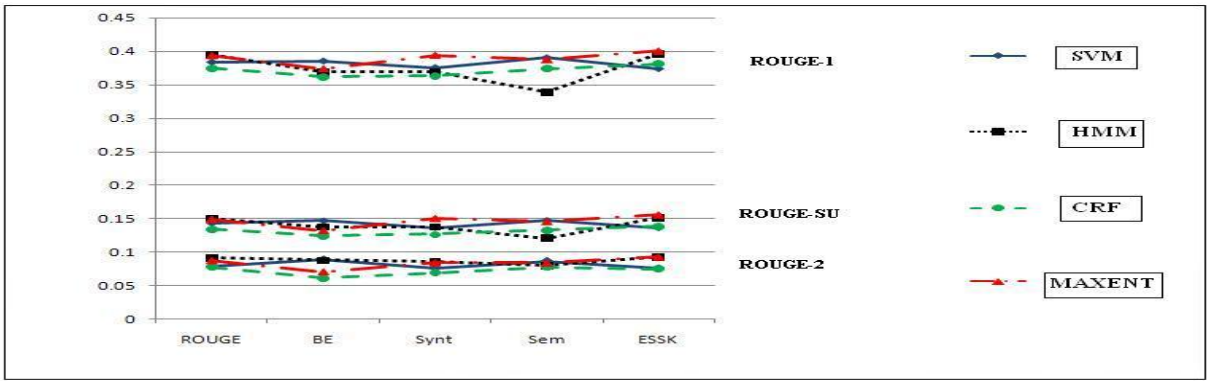

Figure 1: ROUGE F-scores for different supervised systems

best. The other annotation methods' impact is almost similar here in terms of ROUGE-1. Analyzing ROUGE-2 scores, we find that the *BE* performs the best for SVM, on the other hand, *Sem* achieves top ROUGE-SU score. As for the two measures *Sem* annotation is performing the best, we can typically conclude that *Sem* annotation is the most suitable method for the SVM system. *ESSK* works as the best for HMM and *Sem* labeling performs the worst for all ROUGE scores. *Synt* and *BE* labeled HMMs perform almost similar whereas *ROUGE* labeled system is pretty close to that of *ESSK*. Again, we see that the CRF performs best with the *ESSK* annotated data in terms of ROUGE -1 and ROUGE-SU scores and *Sem* has the highest ROUGE-2 score. But *BE* and *Synt* labeling work bad for CRF whereas the *ROUGE* labeling performs decently. So, we can typically conclude that *ESSK* annotation is the best method for the CRF system. Analyzing further, we find that *ESSK* works best for MaxEnt and *BE* labeling is the worst for all ROUGE scores. We can also see that *ROUGE*, *Synt* and *Sem* labeled MaxEnt systems perform almost similar. So, from this discussion we can come to a conclusion that SVM system performs best if the training data uses semantic annotation scheme and *ESSK* works best for HMM, CRF and MaxEnt systems.

4 Conclusion and Future Work

In the work reported in this paper, we have performed an extensive experimental evaluation to show the impact of five automatic annotation methods on the performance of different supervised machine learning techniques in confronting the complex question answering problem. Experimental results show that *Sem* annotation is the best for SVM whereas *ESSK* works well for HMM, CRF and MaxEnt systems. In the near future, we plan to work on finding more sophisticated approaches to effective automatic labeling so that we can experiment with different supervised methods.

References

Jaime Carbonell, Yibing Geng, and Jade Goldstein. 1997. Automated query-relevant summarization and diversity-based reranking. In *IJCAI-97 Workshop on AI in Digital Libraries*, pages 12–19, Japan.

Eugene Charniak. 1999. A Maximum-Entropy-Inspired Parser. In *Technical Report CS-99-12*, Brown University, Computer Science Department.

Michael Collins and Nigel Duffy. 2001. Convolution Kernels for Natural Language. In *Proceedings of Neural Information Processing Systems*, pages 625–632, Vancouver, Canada.

Harold P. Edmundson. 1969. New methods in automatic extracting. *Journal of the ACM*, 16(2):264–285.

Tsutomu Hirao, Jun Suzuki, Hideki Isozaki, and Eisaku Maeda. 2004. Dependency-based sentence alignment for multiple document summarization. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 446–452.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of Workshop on Text Summarization Branches Out, Post-Conference Workshop of Association for Computational Linguistics*, pages 74–81, Barcelona, Spain.

Alessandro Moschitti, Silvia Quarteroni, Roberto Basili, and Suresh Manandhar. 2007. Exploiting Syntactic and Shallow Semantic Kernels for Question/Answer Classificaion. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 776–783, Prague, Czech Republic. ACL.