Abstract

Information about the content structure of a document is largely ignored by current text analysis applications such as information extraction and sentiment analysis. This stands in contrast to the linguistic intuition that rich contextual information should benefit such applications. We present a framework which combines a supervised text analysis application with the induction of latent content structure. Both of these elements are learned jointly using the EM algorithm. The induced content structure is learned from a large unannotated corpus and biased by the underlying text analysis task. We demonstrate that exploiting content structure yields significant improvements over approaches that rely only on local context.

1 Introduction

In this paper, we demonstrate that leveraging document structure significantly benefits text analysis applications. As a motivating example, consider the excerpt from a DVD review shown in Table 1. This review discusses multiple aspects of the product, such as audio and video properties. While the word pleased is a strong indicator of positive sentiment, the sentence in which it appears does not specify the aspect to which it relates. Resolving this ambiguity requires information about global document structure. Most text analysis systems, however, do not exploit such information.

A central challenge in utilizing such information lies in finding a relevant representation of content structure for a specific text analysis task. For instance, when performing single-aspect sentiment analysis, the most relevant aspect of content structure is whether a given sentence is objective or subjective. In a multi-aspect setting, however, information about the sentence topic is required to determine the aspect to which a sentiment-bearing word relates. Therefore, the content structure representation should be intimately tied to a specific text analysis task.

In this work, we present an approach in which a content model is learned jointly with a text analysis task. We assume that we are given complete annotations for the target analysis task; however, the content model component receives only raw, unannotated text. Our approach is implemented in a discriminative framework using latent variables to represent facets of content structure. In this framework, the original task features (e.g., lexical) are conjoined with latent variables to enrich the features with global contextual information. For example, in Table 1, the feature associated with the word pleased should contribute most strongly to

Table 1: An excerpt from a DVD review.

Audio Audio choices are English, Spanish and French Dolby Digital 5.1 ... Bass is still robust and powerful, giving weight to just about any scene – most notably the film’s exciting final fight. Fans should be pleased with the presentation.

Extras This single-disc DVD comes packed in a black amaray case with a glossy slipcover. Cover art has clearly been designed to appeal the Twilight crowd ... Finally, we’ve got a deleted scenes reel. Most of the excised scenes are actually pretty interesting.
the sentiment of the *audio* aspect when it is augmented with a relevant topic indicator.

The coupling of the content model and the task-specific model allows the two components to mutually influence each other during learning. In addition to leveraging unannotated data, our approach allows the content model to be guided by the supervised task data. The combined model can be learned effectively using a novel EM-based method for joint training.

We evaluate our approach on three complementary text analysis tasks. Our first task is a standard information extraction (IE) application where the goal is to populate a predefined template from input texts. We next consider a multi-aspect extractive summarization task in which a system extracts key properties for a pre-specified set of aspects. Finally, we test our model on a multi-aspect sentiment analysis task, where a system predicts the aspect-specific sentiment ratings (Snyder and Barzilay, 2007). On all three tasks, our method for incorporating content structure consistently outperforms structure-unaware counterparts. Moreover, jointly learning content and task parameters yields additional gains over independently learned models.

2 Related Work

Prior research has demonstrated the usefulness of content models for discourse-level tasks. Examples of such tasks include sentence ordering (Barzilay and Lee, 2004; Elsner et al., 2007), extraction-based summarization (Haghighi and Vanderwende, 2009) and text segmentation (Chen et al., 2009). Since these tasks are inherently tied to document structure, a content model is essential to perform them successfully. In contrast, the applications considered in this paper are typically developed without any discourse information. Since these existing models can already effectively capture sentence-level relations, our goal is to augment them with content information rather than to design new text analysis algorithms.

Several applications in information extraction and sentiment analysis are close in spirit to our work (Pang and Lee, 2004; Patwardhan and Riloff, 2007; McDonald et al., 2007). These approaches consider global contextual information when determining whether a given sentence is relevant to the underlying analysis task. For instance, Pang and Lee (2004) refine the accuracy of sentiment analysis by considering only the subjective sentences of a review. McDonald et al. (2007) also assume access to subjectivity annotations, but their method combines subjectivity detection and polarity assessment into a single step. Patwardhan and Riloff (2007) observe similar phenomena in the context of information extraction. Rather than applying their extractor to all the sentences in a document, they limit it to event-relevant sentences. Since they are more likely to contain information of interest, the extraction performance increases.

While our method also incorporates contextual information into existing text analysis applications, our approach is markedly different from these approaches (Pang and Lee, 2004; Patwardhan and Riloff, 2007; McDonald et al., 2007). First, our representation of context encodes more than the relevance-based binary distinction considered in the past work. Our algorithm adjusts the content model dynamically for a given task rather than pre-specifying it. Second, existing approaches assume that relevance information is available during training and therefore content analysis is performed in a supervised fashion. However, relevance annotations are readily available for only a few applications and are prohibitively expensive to obtain for many others. To overcome this drawback, our method induces a content model in an unsupervised fashion and connects it via latent variables to the target model. This design not only eliminates the need for additional annotations, but also allows the algorithm to leverage large quantities of raw data for training the content model. The tight coupling of relevance learning with the target analysis task leads to further performance gains.

Finally, our work relates to machine learning research on labeled topic models (Blei and McAullife, 2007). In this work, latent topic variables are used to generate text as well as a supervised target variable for the document. However, this architecture does not permit the usage of standard discriminative models which condition freely on textual features.
Figure 1: A graphical depiction of the generative process for a labeled document at training time (See Section 3); shaded nodes indicate variables which are observed at training time. First the latent underlying content structure $T$ is drawn. Then, the document text $s$ is drawn conditioned on the content structure utilizing content parameters $\theta$. Finally, the observed information extraction labels for the document are modeled given $s$ and $T$ using the task parameters $\phi$. Note that the arrows for the information extraction labels are undirected since they are modeled discriminatively.

3 Model

In this section, we describe a model which incorporates content information into an information extraction (IE) task.\(^1\) Our approach assumes that at training time we have a collection of labeled documents $D_L$, each consisting of the document text $s$ and true task-specific labeling $y^*$. For the information extraction task, $y^*$ consists of sequence labels for the tokens of a document. Specifically, the document text $s$ is composed of sentences $s_1, \ldots, s_n$ and the labelings $y^*$ consists of corresponding label sequences $y_1, \ldots, y_n$.\(^2\)

As is standard in most information extraction work, we model each $y_i$ using a CRF which conditions on the observed document text. In this work, we also assume a content model, which we fix to be the document-level HMM as used in Barzilay and Lee (2004). In this content model, each sentence $s_i$ is associated with a hidden topic variable $T_i$ which generates the words of the sentence. We will use $T = (T_1, \ldots, T_n)$ to refer to the hidden topic sequence for a document. We fix the number of topics to a pre-specified constant $K$.

A simple way to incorporate the content model with the CRF task model would be to independently induce the document-level HMM and use its predictions about sentence topics as additional features in a conditional random field (CRF) (Lafferty et al., 2001). For instance, using the example from Table 1, we could have a feature that indicates the word “pleased” conjoined with the segment topic. These topic-specific features serve to disambiguate word usage. While this approach allows document structure to influence text analysis, there is no guarantee the structure learned by the content model corresponds to one which is useful for the text analysis task. Essentially, we need to close the loop and allow the task to influence the learned content representation.

Our approach facilitates this joint learning by incorporating both content and task components into a unified model. This model, depicted in Figure 2, proceeds as follows: First the document-level HMM

\(^1\)In Section 3.4, we discuss how this framework can be used for other text analysis applications.

\(^2\)Note that each $y_i$ is a label sequence across the words in $s_i$, rather than an individual label.
generates a hidden content topic sequence $T$ for the sentences of a document. This content component is parametrized by $\theta$ and decomposes in the standard HMM fashion:

$$P_\theta(s, T) = \prod_{i=1}^{n} P_\theta(T_i|T_{i-1}) \prod_{w \in s_i} P_\theta(w|T_i) \quad (1)$$

Then the label sequences for each sentence in the document are independently modeled as CRFs which condition on both the sentence features and the sentence topic:

$$P_\phi(y|s, T) = \prod_{i=1}^{n} P_\phi(y_i|s_i, T_i) \quad (2)$$

Each sentence CRF is parametrized by $\phi$ and takes the standard form,

$$P_\phi(y|s, T) \propto \exp \left\{ \sum_j \phi^T_j \left[ f_N(y^j, s, T) + f_E(y^j, y^{j+1}) \right] \right\}$$

where $f_N(\cdot)$ and $f_E(\cdot)$ are feature functions associated with CRF nodes and transitions respectively.

This joint process, depicted graphically in Figure 1, is summarized as,

$$P(T, s, y^*) = P_\theta(T, s)P_\phi(y^*|T, s) \quad (3)$$

Note that this probability decomposes into a document-level HMM term (the content component) as well as a product of CRF terms (the information extraction component).

### 3.1 Learning

During learning, we would like to find the document-level HMM parameters $\theta$ and the IE task CRF parameters $\phi$ which maximize the likelihood of the labeled documents. The only observed elements of a labeled document are the document text $s$ and the IE labels $y^*$. This objective is given by:

$$L_L(\phi, \theta) = \sum_{(s, y^*) \in D_L} \log P(s, y^*)$$

$$= \sum_{(s, y^*) \in D_L} \sum_{T} P(T, s, y^*)$$

We use the EM algorithm to optimize this objective.

### E-Step

The E-Step in EM requires computing the posterior distribution over latent variables. In this model, the only latent variables are the sentence topics $T$. To compute this term, we utilize the decomposition in Equation (3) and rearrange HMM and CRF terms to obtain:

$$P(T, s, y^*) = P_\theta(T, s)P_\phi(y^*|T, s)$$

$$= \left( \prod_{i=1}^{n} P_\theta(T_i|T_{i-1}) \prod_{w \in s_i} P_\theta(w|T_i) \right)$$

$$\left( \prod_{i=1}^{n} P_\phi(y^*_i|s_i, T_i) \right)$$

$$= \prod_{i=1}^{n} P_\theta(T_i|T_{i-1}) \left( \prod_{w \in s_i} P_\theta(w|T_i)P_\phi(y^*_i|s_i, T_i) \right)$$

We note that this expression takes the same form as the document-level HMM, except that in addition to emitting the words of a sentence, we also have an observation associated with the sentence sequence labeling. We treat each $P_\phi(y^*_i|s_i, T_i)$ as part of the node potential associated with the document-level HMM. We utilize the Forward-Backward algorithm as one would with the document-level HMM in isolation, except that each node potential incorporates this CRF term.

### M-Step

We perform separate M-Steps for content and task parameters. The M-Step for the topic parameters is identical to the document-level HMM content model: topic emission and transition distributions are updated with expected counts derived from E-Step topic posteriors.

The M-Step for the task parameters does not have a closed-form solution. Recall, that in the M-Step, we maximize the log probability of all random variables given expectations of latent variables. Using the decomposition in Equation (3), it is clear the only component of the joint labeled document probability which relies upon the task parameters is $\log P_\phi(y^*|s, T)$. Thus for the M-Step, it is sufficient...
to optimize the following with respect to $\phi$:

$$
\mathbb{E}_{T|s, y^*} \log P_\phi(y^*|s, T) = \sum_{i=1}^{n} \mathbb{E}_{T|s_i, y_i^*} \log P_\phi(y_i^*|s_i, T_i) = \sum_{i=1}^{n} \sum_{k=1}^{K} P(T_i = k|s_i, y_i^*) \log P_\phi(y_i^*|s_i, T_i)
$$

The first equality follows from the decomposition of the task component into independent CRFs (see Equation (2)). Optimizing this objective is equivalent to a weighted version of the conditional likelihood objective used to train the CRF in isolation. An intuitive explanation of this process is that there are multiple CRF instances, one for each possible hidden topic $T$. Each utilizes different content features to explain the sentence sequence labeling. These instances are weighted according to the posterior over $T$ obtained during the E-Step. While this objective is non-convex due to the summation over $T$, we can still optimize it using any gradient-based optimization solver; in our experiments, we used the LBFGS algorithm (Liu et al., 1989).

### 3.2 Inference

We must predict a label sequence $y$ for each sentence $s$ of the document. We assume a loss function over a sequence labeling $y$ and a proposed labeling $\hat{y}$, which decomposes as,

$$
L(y, \hat{y}) = \sum_{j} L(y^j, \hat{y}^j)
$$

where each position loss is sensitive to the kind of error which is made. Not extracting a token is penalized to a greater extent than extracting it with the wrong label:

$$
L(y^j, \hat{y}^j) = \begin{cases} 
0 & \text{if } \hat{y}^j = y^j \\
1 & \text{if } y^j = \text{NONE and } \hat{y}^j \neq y^j \\
c & \text{if } y^j \neq \text{NONE and } \hat{y}^j \neq y^j
\end{cases}
$$

In this definition, NONE represents the background IE label which are reserved for tokens which do not correspond to labels of interest. The constant $c$ represents a user-defined trade-off between precision and recall errors. For our information extraction and multi-aspect summarization task, we select $c = 5$ to combat the high-precision bias typical of conditional likelihood models.

At inference time, we select the single labeling which minimizes the expected loss with respect to model posterior over label sequences:

$$
\hat{y} = \min_{\hat{y}} \mathbb{E}_{y|s} L(y, \hat{y}) = \min_{\hat{y}} \sum_{j=1}^{K} \mathbb{E}_{y|s} L(y^j, \hat{y}^j)
$$

In our case, we must marginalize out the sentence topic $T$:

$$
P(y^j|s) = \sum_{T} P(y^j, T|s) = \sum_{T} P_\theta(T|s) P_\phi(y^j|s, T)
$$

This minimum risk criterion has been widely used in NLP applications such as parsing (Goodman, 1999) and machine translation (DeNero et al., 2009). Note that the above formulation differs from the standard CRF due to the latent topic variables. Otherwise the inference task could be accomplished by directly obtaining posteriors over each $y^j$ state using the Forward-Backwards algorithm on the sentence CRF.

Finding $\hat{y}$ can be done efficiently. First, we obtain marginal token posteriors as above. Then, the expected loss of a token prediction is computed as follows,

$$
\sum_{\hat{y}^j} P(y^j|s) L(y^j, \hat{y}^j)
$$

Once we obtain expected losses of each token prediction, we compute the minimum risk sequence labeling by running the Viterbi algorithm. The potential for each position and prediction is given by the negative expected loss. The maximal scoring sequence according to these potentials, minimizes the expected risk.

### 3.3 Leveraging unannotated data

Our model allows us to incorporate unlabeled documents, denoted $D_U$, to improve the learning of the content model. For an unlabeled document we only
observe the document text $s$ and assume it is drawn from the same content model as our labeled documents. The objective presented in Section 3.1 assumed that all documents were labeled; here we supplement this objective which also captures the likelihood of unlabeled documents according to the content model:

$$L_U(\theta) = \sum_{s \in \mathcal{D}_L} \log P_\theta(s)$$

$$= \sum_{s \in \mathcal{D}_L} \log \sum_{T} P_\theta(s, T)$$

Our overall objective function is to maximize the likelihood of both our labeled and unlabeled data. This objective corresponds to,

$$L(\phi, \theta) = L_U(\theta) + L_L(\phi, \theta)$$

This objective can also be optimized using the EM algorithm, where the E-Step for labeled and unlabeled documents is outlined above.

### 3.4 Generalization

The approach outlined can be applied to a wider range of task components. For instance in Section 4, we apply this approach to a regression task. The model structure still decomposes as in Figure 1, the details of learning are slightly different. For instance, because the task label (here a sentiment rating) is not localized in any region of the document, all content model variables influence the target response. Conditioned on the target label, all topic variables become correlated, requiring a summation over all topic assignments. For the case of our multi-aspect sentiment task, this computation can be done exactly. If summation is intractable, the posterior may be approximated using either Gibbs sampling or variational techniques.

Note that our third task, multi-aspect summarization, uses the same formulation as the information extraction task.

### 4 Experimental Set-Up

We apply our approach to three text analysis tasks that stand to benefit from modeling content structure: information extraction, multi-aspect sentiment analysis and multi-aspect review summarization.
**Information Extraction** We consider a standard information extraction set-up, where the goal is to populate a pre-defined template with phrases extracted from input documents. This task has been extensively studied in prior work, and is commonly modeled using a chain CRF over sentence words, wherein states correspond to extraction labels (Lafferty et al., 2001). Traditional information extraction models rely primarily on lexical features such as word identity, capitalization properties and numerical properties.

For this task, we use the Acquisitions corpus, a benchmark data set commonly used for evaluating IE systems (Lewis, 1992). The corpus consists of short articles about corporate mergers and acquisitions (see Table 2 for relevant statistics). The annotation scheme identifies 13 categories of interest such as purchaser, seller, dollar amount and acquisition location.

Prior IE research has demonstrated that incorporating sentence relevance improves extraction accuracy (Patwardhan and Riloff, 2007). Since we rely on the content model to make this distinction, our HMM-based model has only two topics, intended to model sentence relevance.

**Multi-Aspect Sentiment Ranking** The goal of multi-aspect sentiment classification is to predict a set of numeric ranks that reflects the user satisfaction for each aspect (Snyder and Barzilay, 2007). One of the challenges in this task is to attribute sentiment-bearing words to the aspects they describe. Information about document structure has potential to greatly reduce this ambiguity.

Following standard sentiment ranking approaches (Wilson et al., 2004; Pang and Lee, 2005; Goldberg and Zhu, 2006; Snyder and Barzilay, 2007), we employ ordinary linear regression to independently map bag-of-words representations into predicted aspect ranks. In addition to commonly used lexical features, this set is augmented with content features as described above. For this application, we fix the number of HMM states to be equal to the predefined number of aspects.

We test our sentiment ranker on a set of DVD reviews from the website IGN.com. Each review is accompanied by 1-10 scale ratings in four categories that assess the quality of a movie’s content, video, audio, and DVD extras. In this data set, segments corresponding to each of the aspects are clearly delineated in each document. Therefore, we can compare the performance of the algorithm using automatically induced content models against the gold standard structural information.

**Multi-Aspect Review Summarization** The goal of this task is to extract informative phrases that identify properties of products for each aspect of interest. Variants of this task have been considered in review summarization in previous work (Branavan et al., 2009; Kim and Hovy, 2006). See Figure 3a for a concrete example with annotated phrases for multiple aspects. This task has elements of both information extraction and phrase-based summarization — it is broader in scope than standard template-driven IE, but at the same time more constrained than generic extractive summarization. The difficulty here is that phrase selection is highly context-dependent. For instance, in Figure 3a, the highlighted phrase `easy-to-use` might refer to either the menu or remote; broader context is required for correct labeling.

As in the IE task, we employ a CRF where states correspond to aspects of interest. We fix the number of topics to be equal to the number of aspects.

We evaluate our techniques on a corpus of Amazon TV reviews (see Table 2 for details). To eliminate noisy reviews, we only retain documents that have been rated “helpful” by the users of the site. See Figure 3a for example review excerpts and a list of aspects used. The labeled training and test data sets were annotated manually with aspect labels. To test agreement, two native speakers annotated a set of four documents. The agreement between the judges was 0.54 as measured by Cohen’s Kappa. The rest of the documents were annotated using Mechanical Turk. Since we cannot select high-quality annotators directly, we include a control document with the documents assigned to each annotator. The work of any annotator who exhibits low agreement on the control document annotation is excluded from the corpus.

**4.2 Baseline Comparison and Evaluation**

**Baselines** For all the models, we obtain a baseline system by eliminating content features and only us-
Table 2: This table summarizes the size of each corpus. In each case, the unlabeled texts of both labeled and unlabeled documents are used for training the content model, while only the labeled training corpus is used to train the task model.

| Task                          | Labeled | Unlabeled | Avg. Size |
|-------------------------------|---------|-----------|-----------|
|                               | Train   | Test      | Words     | Sents    |
| Multi-aspect sentiment        | 600     | 65        | 0         | 1,027    | 20.5     |
| Multi-aspect summarization    | 14      | 21        | 12,684    | 214      | 11.7     |
| Information extraction        | 250     | 300       | 3,459     | 197      | 9.3      |

Evaluation Metrics For the multi-aspect summarization and information extraction tasks, we measure token precision and recall of label assignments. As commonly used in information extraction, we also employ binary F-measure, a coarser metric which measures extraction precision and recall (i.e., ignoring labels). For the summarization task, we also report ROUGE (Lin, 2004). We compute ROUGE by comparing system output with a labeled document. To control for length, we limit each system to predict the same number of tokens as the original labeled document.

For multi-aspect sentiment ranking, we report the average $L_2$ (squared difference) and $L_1$ (absolute difference) between system prediction and true 1-10 sentiment rating across test documents and aspects.

Our metrics of statistical significance vary by task. For the sentiment task, we use Student’s t-test. For the multi-aspect summarization task, we perform chi-square analysis on the ROUGE scores. For the IE task, we perform chi-square analysis on precision and recall separately, as is common in that field (Finkel and Manning, 2009).

Table 3: The error rate on the multi-aspect sentiment ranking (the maximal error is 10). We report mean $L_1$ and $L_2$ between system prediction and true values over all aspects. Results marked with * are statistically significant with $p < 0.05$.

|        | $L_1$ | $L_2$ |
|--------|-------|-------|
| NoCM   | 1.37  | 3.15  |
| IndepCM| 1.28* | 2.80* |
| JointCM| 1.25* | 2.65* |
| Gold   | 1.18  | 2.48  |

Figure 4: Results using half of the annotated documents with varying amounts of unlabeled data on the summarization task. The horizontal line represents the NoCM baseline using the full annotated set.

5 Results

In this section, we present the results of the methods on the three tasks described above (see Tables 3, 4, and 5). In all the three applications, the baseline numbers are comparable to existing systems on tasks of similar complexity.

Baseline Comparisons The JointCM model outperforms the NoCM baseline consistently on all three tasks. The highest error reduction — 12.5% — is achieved on multi-aspect summarization, fol-
Table 4: Results on information extraction. Precision and recall marked with * are statistically significant with $p < 0.05$.

![Figure 5: The accuracy of multi-aspect summarizer as a function of raw data for content model training.](image)

- The quality of the induced content model is determined by the amount of training data. As Figure 5 shows that the multi-aspect summarizer improves with the increase in the size of raw data available for learning content model.

Compensating for annotation sparsity We hypothesize that by incorporating rich contextual information, we can reduce the need for manual task annotation. We test this hypothesis by reducing the amount of annotated data available to the method, while increasing the amount of unannotated data. As Figure 4 shows, the performance increase achieved by doubling annotated data can also be achieved in our model by adding only 3,000 documents of raw data.

6 Conclusion

This paper demonstrated the benefits of incorporating content models in text analysis tasks. Our results empirically connect model quality and task performance, suggesting that further improvements in content modeling may yield even further gains.
References

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In Proceedings of the NAACL/HLT, pages 113–120.

David M. Blei and Jon D. McAuliffe. 2007. Supervised Topic Models. In NIPS.

S. R. K. Branavan, Harr Chen, Jacob Eisenstein, and Regina Barzilay. 2009. Learning document-level semantic properties from free-text annotations. JAIR, 34:569–603.

Harr Chen, S. R. K. Branavan, Regina Barzilay, and David R. Karger. 2009. Content modeling using latent permutations. JAIR, 36:129–163.

John DeNero, David Chiang, and Kevin Knight. 2009. Fast consensus decoding over translation forests. In Proceedings of the ACL/IJCNLP, pages 567–575.

Micha Elsner, Joseph Austerweil, and Eugene Charniak. 2007. A unified local and global model for discourse coherence. In Proceedings of the NAACL/HLT, pages 436–443.

Jenny Rose Finkel and Christopher D. Manning. 2009. Joint parsing and named entity recognition. In Proceedings of the NAACL.

Andrew B. Goldberg and Xiaojin Zhu. 2006. Seeing stars when there aren’t many stars: Graph-based semi-supervised learning for sentiment categorization. In Proceedings of the NAACL/HLT Workshop on TextGraphs, pages 45–52.

Joshua Goodman. 1999. Semiring parsing. Computational Linguistics, 25(4):573–605.

Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In Proceedings of the NAACL/HLT, pages 362–370.

Soo-Min Kim and Eduard Hovy. 2006. Automatic identification of pro and con reasons in online reviews. In Proceedings of the COLING/ACL on Main conference poster sessions, pages 483–490.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of ICML, pages 282–289.

David Dolan Lewis. 1992. Representation and learning in information retrieval. Ph.D. thesis, University of Massachusetts.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In Proceedings of the ACL, pages 74–81.

Dong C. Liu, Jorge Nocedal, Dong C. Liu, and Jorge Nocedal. 1989. On the limited memory bfgs method for large scale optimization. Mathematical Programming, 45:503–528.