Text Structure-Aware Classification

by

Zoran Dzunic

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2009

Author .....
Department of Electrical Engineering and Computer Science
February 11, 2009

Certified by .....
Regina Barzilay
Associate Professor
Thesis Supervisor

Accepted by .....
Terry P. Orlando
Chairman, Department Committee on Graduate Students

Text Structure-Aware Classification

by

Zoran Dzunic

Submitted to the Department of Electrical Engineering and Computer Science
on January 30, 2009, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

Bag-of-words representations are used in many NLP applications, such as text classification and sentiment analysis. These representations ignore relations across different sentences in a text and disregard the underlying structure of documents. In this work, we present a method for text classification that takes into account document structure and only considers segments that contain information relevant for a classification task. In contrast to the previous work, which assumes that relevance annotation is given, we perform the relevance prediction in an unsupervised fashion. We develop a Conditional Bayesian Network model that incorporates relevance as a hidden variable of a target classifier. Relevance and label predictions are performed jointly, optimizing the relevance component for the best result of the target classifier. Our work demonstrates that incorporating structural information in document analysis yields significant performance gains over bag-of-words approaches on some NLP tasks.

Thesis Supervisor: Regina Barzilay
Title: Associate Professor

Acknowledgments

I owe deep gratitude to my advisor, Regina Barzilay, for the great guidance and support that she has been tirelessly providing to me. My work is founded on her ideas. I acknowledge Amir Globerson for his huge contribution to the project and many discussions that we had, and Professor Martin Rinard, whose high spirit and valuable advices kept me on my track.

I also thank Yoong Keok Lee for the productive collaboration in the early stages of the project, Branavan, who helped me on many occasions, Serdar Balci, my roommate and friend, Benjamin Snyder, Tahira Naseem, Christina Sauper, Erdong Chen, Harr Chen, Jacob Eisenstein, Pawan Deshpande, Igor Malioutov, Viktor Kuncak, Karen Zee, Michael Carbin, Patrick Lam, Darko Marinov, and other colleagues for their friendship and support.

My work would not be accomplished without my family: My mother, who is my biggest fan, my father, my aunt, my niece and my nephew.

Finally, I am dedicating this thesis to my dearest Ivana, who was beside me day and night, whether it was good or bad, to cheer me up and make me enjoy every moment.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Today, most approaches to text classification are based on the bag-of-words representation where a document is mapped to a vector of word counts [14, 15]. While easy to compute and manipulate, this representation ignores relations across different sentences in text and disregards the underlying structure of a document. The oversimplicity of this abstraction limits the scope of the problems that text classification can address and reduces accuracy. The problem can be particularly acute when processing long documents which intertwine several topics.

Consider, for instance, the extract from a Reuters article shown in Table 1.1. While the first segment of this story describes a recent event, the last segment describes previous activities used by the Maoist rebels. If our task is to classify the type of terrorist event, the words from the last segment would confuse a typical bag-of-words classifier. Ideally, we would like to apply a classifier only to the segments which contain information relevant for the target classification task. Unfortunately, we cannot make this distinction in the bag-of-words framework since it does not preserve information pertaining to document structure.

If we know ahead of time which segments contain information of interest, we can employ a target classifier to process only these relevant segments. This approach has been successfully implemented in sentiment classification where only subjective segments are considered to be relevant [23, 19]. However, in many applications information about relevance cannot be easily obtained. Moreover, it can be challenging, even for humans, to predict what information is relevant for an automatic classifier.

Two unidentified gunmen killed Ramesh Manandhar, a security guard with the American embassy in the Nepali capital on Saturday. He was killed near the compound of the U.S. Agency for International Development (USAID) ... Maoist rebels, who are fighting to install a one-party communist republic in the kingdom, broke a four -month-old truce in November. Last month they also bombed a Coca-Cola plant in Kathmandu.

Table 1.1: An excerpt from Reuters' article.

In this work we present an approach that takes into account segment relevance but does not require any additional annotations. We model relevance as a hidden variable of a target classifier, so that relevance assessment and target classification are performed simultaneously. This formulation allows us to learn relevance in a task-specific fashion optimized for the final classification performance. Every segment is associated with a hidden variable, which value determines whether features of that segment contribute to the target classification. Furthermore, by defining additional features on the hidden variables, we can encode our intuition about the properties that contribute to segment relevance prediction.

We develop two implementations of our model. The first implementation formally corresponds to a conditional Bayesian network (CBN), since it posits that the predicted label is a result of a sequence of decisions, each modeled by a conditional distribution [25]. The first decision is about the relevance of each segment, and the second decision uses the relevance judgments to infer the label of the document. Each conditional distribution is modeled with a log-linear classifier [1, 27].

The second implementation corresponds to a conditional random field (CRF) with hidden variables [26]. It is globally normalized undirected version of CBN, which directly models the decision about the label given the document as a conditional distribution using log-linear classifier. Each intermediate decision above is incorporated through a factor in the model, but its conditional distribution is not modeled explicitly.

The two implementations produce similar results on classification tasks. However, there

are several differences. The CBN implementation might explain better the label assignment through relevance assignments, since we model the probability of each segment relevance directly. On the other hand, the lack of local normalization in the CRF implementation makes exact training and inference possible, while we use Gibbs sampling for approximate training and inference in the CBN implementation. We also explore several schemes for feature integration over relevant segments. Our default scheme is to average features across relevant segments, which enables efficient inference in the CRF implementation using path aggregation [22].

We evaluate our model on two tasks: topical classification and polarity classification. We create a topical classification dataset using the MIPT terrorism knowledgebase [1], where a label corresponds to a tactic used in a terrorist attack. Each document is created by combining a paragraph about the event of interest with several paragraphs from the background information on the terrorist organization, in order to simulate a typical structure of a news article. For polarity classification, we use polarity dataset of movie reviews created by Pang and Lee [23]. We experiment with two levels of granularity: paragraph level and sentence level. Standard bag-of-words log-linear model is a baseline that corresponds to our model. We also include SVM baseline [10] in the comparison. The two baselines produce similar results, as expected.

The outcome of our experiments is different on these two tasks. While we show that the introduction of relevance structure improves topical classification accuracy, we do not see a benefit of relevance modeling on the polarity classification task. In conclusion, the utilization of our method may strongly depend on the properties of a dataset.

We demonstrate that our relevance-aware model yields significant gains in comparison to a standard bag-of-words approach on the topical classification task in the case of paragraph level granularity. Due to the way the data is constructed, we know which paragraphs are relevant. We create an oracle baseline that accesses this information and use only relevant paragraphs. The results of our method match the results of the oracle baseline, although our model identifies roughly two times more paragraphs as relevant comparing to the number of relevant segments given by construction. Manual inspection shows that

[1] http://www.tkb.org/

our method eliminates paragraphs in a conservative way – it does not eliminate neutral paragraphs (that do not affect the classification outcome). On the other hand, it eliminates paragraphs that can confuse the classifier.

In a further experiment, we show that the classification improvement stems from the structure of our model, rather than from the extended feature set. Namely, we use unigram feature set to predict the final label, and extend it with several features (such as verb tenses, etc.) to predict relevance. However, even if we use only unigrams for relevance prediction, the result does not change significantly. Also, when we use the extended feature set in conjunction with the baseline, the results do not change, therefore, showing that simple addition of features is not sufficient.

Finally, we observe that our method drops the edge over the baseline when we switch to the sentence level granularity. This can be explained by the fact that a sentence does not contain enough information for the relevance prediction.

On a polarity classification task, our model does not improve over the baseline. We observe that it gives best result when the relevance prediction parameters are regularized heavily, such that they practically become zero. This leads to equal relevance of all segments, and, therefore, the relevance-aware model simply recreates the baseline. By reducing such an extreme penalization, we see that the model do not learn to predict relevance properly, and the classification accuracy decreases. We conclude that there is no enough pattern in the data to predict relevance in a way that is consistent with the final classification task. [2] It is also worth noting that reviews tend to be entirely positive or negative, i.e., users who like/dislike a product tend to write only positive/negative comments about it. Therefore, there might be enough evidence for the correct polarity classification even when looking at the whole document (in a positive review, positive comments will outweight the negative ones, and vice versa). As a supporting fact, Pang and Lee [23] also do not show improvement over the SVM baseline on the same dataset, even though they train relevance model in a supervised fashion. McDonald et al. [19] get only small improvement, and it is

[2]Note that, since we model relevance in an unsupervised fashion, we do not relate relevance to the notion of subjectivity exploited in the previous work, but rather to the predictability for the final classification task.

not consistent across different datasets. [3]

The remainder of thesis is organized in the following way. In the next chapter, we discuss related work on document categorization. The work of Pang and Lee [23], McDonald et al. [19], and Patwardhan and Riloff [24] are the most influential on our work, and we discuss them in more detail. In Chapter 3, we provide a formal description of our relevance-aware model and its two implementations. In Chapter 4, we describe the experimental set-up, the details of the training procedure, and the results of our model (both implementations) on the two tasks. Finally, we make final remarks and give directions for future research in Chapter 5.

[3] They use different datasets.

Chapter 2

Related Work

Text classification is a popular research topic in natural language processing, information retrieval and machine learning. A large body of work in this area aims to improve classification performance by refining document representation. These attempts include enriching the bag-of-words representation with contextual, syntactic and semantic information [30, 29, 21], often using Wordnet as a resource [20, 8].

The work presented in this paper explores an orthogonal source of knowledge — document structure. Our approach is motivated by empirical results that demonstrate the benefits of relevance-aware processing across several classification applications. For instance, Pang and Lee [23] refine the accuracy of sentiment analysis by considering only subjective sentences of a review. McDonald et al. [19] also assume access to subjectivity annotations, but their method combines subjectivity detection and polarity assessment into a single step. Patwardhan and Riloff [24] 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 segments. Since these segments are more likely to contain information of interest, the extraction performance increases.

While our method is also driven by the notion of relevance, our model is markedly different from previous approaches [23, 24, 19]. These approaches assume that information about segment relevance is available during training and therefore relevance classification is performed in a supervised fashion. However, relevance annotations are readily available only in a few cases and would be prohibitively expensive for others. To overcome this

drawback, our method models relevance as a latent variable in the target classifier, eliminating the need for additional annotations. The tight coupling of relevance learning with the target classification leads to further performance gains.

Discriminative latent variable models were previously used by Quattoni et al. [26] who extended the conditional random field (CRF) framework [12] to incorporate hidden variables. While the original model was developed in the context of object recognition [26], the variants of the hidden variable CRF have been applied to several language processing tasks, including parsing [11] and co-reference resolution [6]. Here we introduce a variant on this framework, which explicitly models the two step classification procedure that arises in relevance modeling. It thus uses a conditional Bayesian network instead of the conditional random fields employed in [26].

In the sequel of this Chapter, we describe the work of Pang and Lee [23], McDonald et al. [19], and Patwardhan and Riloff [24] in more detail.

2.1 Cascaded Models for Sentiment Analysis

Pang and Lee [23] explore subjectivity summarization and apply it to polarity classification of movie reviews. The assertion is that the polarity of a review is determined from its subjective part. In other words, subjective portion is relevant for classification.

2.1.1 Problem Description

Polarity classification is a popular task in sentiment analysis, in which reviews are labeled "thumbs up" or "thumbs down". Standard bag-of-words classifiers can be employed, where feature vector indicates presence of words from a defined vocabulary. [1]

Some words that influence the classification significantly, such as "bad", "good", etc., may be used in the parts of a review where sentiment information is not present. For example, a phrase "bad guys" can refer to the characters of a movie. If it appears in the

[1] Typically, vocabulary is built from training corpus, eliminating words that appear fewer times than some threshold

plot description of a positive review, it may confuse the classifier to classify it wrongly as negative.

In order to reduce confusion, only subjective part of a review (subjective summary) is passed as an input to a polarity classifier.

2.1.2 Method

Pang and Lee use a two-step approach. A **subjectivity classifier** is applied to each sentence of a full review. Subjective sentences form a subjective summary that serves as an input to a **polarity classifier**.[2] Naïve Bayes [13] and SVM [10] are used as base classifiers in both steps.

In addition, they improve the subjectivity detection by applying a cut-based method to determine the subjectivity of all sentences simultaneously. In this method, sentences are split into two classes: subjective (Sub) and objective (Obj). $ind_C(s_i)$ represents individual score of sentence s_i affiliation to class $C, C \in \{Sub, Obj\}$. $assoc(s_i, s_j)$ represents association score of sentences s_i and s_j affiliation to the same class. All scores are non-negative values and represent an input to a cut-based method. Given these definitions, the method partitions sentences into two classes such that the following expression is maximized:

$$\sum_{s \in Sub} ind_{Sub}(s) + \sum_{s \in Obj} ind_{Obj}(s) + \sum_{s_i, s_j \in Sub} assoc(s_i, s_j) + \sum_{s_i, s_j \in Obj} assoc(s_i, s_j), \quad (2.1)$$

which is equivalent to minimizing the expression

$$\sum_{s \in Sub} ind_{Obj}(s) + \sum_{s \in Obj} ind_{Sub}(s) + \sum_{s_i \in Sub, s_j \in Obj} assoc(s_i, s_j). \quad (2.2)$$

This formulation is equivalent to a min-cut problem in the graph in Figure 2-1. Each sentence is represented with a node. There are two additional nodes representing the two classes (Sub and Obj). Individual scores are attached to edges that connect a class node to a sentence node, while association scores are attached to edges connecting two sentence

[2] at both training and test time

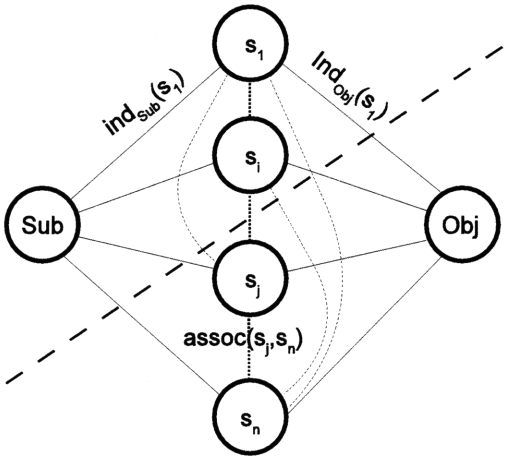

Figure 2-1: Graph Representation of Cut-Based Subjectivity Detection Method. Nodes Sub and Obj represent sets of subjective and objective sentences, respectively. Nodes $s_1, \ldots, s_n$ represent sentences.

nodes.

Advantages of cut-based method are: its efficiency (finding a min-cut in a graph requires polynomial running time using maximum-flow algorithms) and flexibility in computing scores associated with edges. Pang and Lee set individual scores of a sentence based on the output of a subjectivity classifier on that sentence. In the case of Naïve Bayes classifier, $ind_{Sub}(s) = P_{sub}(s)$, and $ind_{Obj}(s) = 1 - P_{sub}(s)$, where $P_{sub}(s)$ is a probability of a sentence being subjective. In the case of SVM, weight produced by SVM (signed distance to the separating hyperplane) is converted to an individual score. Association score between two sentences is set based on their proximity. Authors try several different non-increasing functions of the distance between sentences and choose the one that gives the best performance on polarity classification task.[3]

2.1.3 Resuts

Polarity dataset consists of 1000 positive and 1000 negative movie reviews. Pang and Lee mine the web to create a subjectivity dataset. It is a collection of 5000 subjective review

[3] Subjectivity corpus consists of individual sentences coming from different reviews, and therefore cannot be used for the evaluation of the cut-based method.

	NB	SVM
NB	86.4	86.4
NB+PROX	86.6	86.5
SVM	85.2	85.45
SVM+PROX	86.4	86.15
FULL REVIEW	82.8	87.15

Table 2.1: Results presented by Pang&Lee. Rows correspond to different subjectivity extraction methods, while columns correspond to polarity classification methods. PROX indicates the use of the cut-based method.

snippets and 5000 objective sentences from plot summaries. These two datasets are from the same domain, but do not overlap in terms of movies they describe.

Table 2.1 shows a summary of results presented in [23].[4] In the case of Naïve Bayes polarity classifier, classification based on a subjective summary significantly outperforms classification based on a full review. It is not the case with SVM as a polarity classifier, when applying subjectivity detection leads to slightly worse, but statistically not different results. However, these results are obtained using only a portion of a document (about 60%). From that perspective, authors conclude that the subjective extracts preserve the sentiment information in the original documents, and thus are good summaries from the polarity-classification point of view. Furthermore, they investigate classification accuracy as a function of the number of sentences included in the summary.[5] They show that the result using 15-sentence summaries is close to the full review result. Also, they show that such summaries are more informative than standard document summaries in terms of polarity classification.

Finally, Pang and Lee show that applying cut-based method to subjectivity detection leads to improved results on polarity classification (Table 2.1). Improvements are statistically significant in the case of SVM subjectivity classifiers.

[4]Some numbers in the table are explicitly given in the paper, while some of them are read from the graph.
[5]Sentences are added in decreasing order, based on the output of subjectivity classifier.

2.1.4 Comparison to Our Method

The main difference from our work is that Pang and Lee train subjectivity and polarity classifiers separately, both in full supervision, whereas we define a joint model in which relevance is incorporated in an unsupervised fashion. Therefore, our method does not require any subjectivity-labeled data. Also, since we tune the relevance part of our model for the best performance on the final classification task, what we recover as relevant may not coincide with what is truly subjective. Relevant part is what best helps the classification task.

2.2 Structured Models for Sentiment Analysis

McDonald et al. [19] develop a structured model for joint sentiment classification at different levels of granularity within a document. They apply their method to three reviews datasets in different domains and show that it outperforms individual document and sentence level models, as well as a cascaded model in which sentiment information is passed only in one direction – from sentence to document level. Therefore, there is a benefit of joint modeling, in which sentiment information is passed in both directions.

2.2.1 Motivation

Sentiment analysis is an important yet challenging task in Natural Language Processing and Information Retrieval. Different application needs guided work on sentiment extraction on different levels of granularity. While question answering system may only require document sentiment label, a summarization task would probably benefit from sentence level sentiment information.

McDonald et al. argue that sentiment information on one level can be beneficial for sentiment analysis on another level. For example, look at the following review excerpt:[6]
This is the first Mp3 player that I have used ...I thought it sounded great ...After only a few weeks, it started having trouble with the earphone connection ...I won't be buying

[6]Both examples are taken from [19].

another.

Although this review says something positive about the product, it is overall negative. However, a document level classifier might make a mistake because of the presence of the strong positive word *great*. This would not be the case if sentence level sentiment information is present and there are some ties between the two levels. For example it is typical that the sentiment of the last sentence coincides with the sentiment of the whole document. If that relationship is captured, document would probably be classified correctly. In another example the information is passed in the opposite direction:

My 11 year old daughter has also been using it and it is a lot harder than it looks.

Although this sentence in isolation would most likely be labeled as negative, it is a part of a positive review where *harder* refers to a "good workout" on a fitness equipment. Therefore, the overall document sentiment may help disambiguate the meaning of "hard".

These two examples justify the approach of McDonald et al., in which sentiment information flows in both directions. In that sense, their model is superior in comparison to the cascaded model of Pang and Lee [23].

2.2.2 Method

McDonald et al. introduce sentence-document model in Figure 2-2. It is an undirected graphical model in which each sentence label depends on the previous and next sentence labels, as well as on the document label, while the document label depends on the labels of all sentences. All labels depend on the observed document $\mathbf{s}$, (represented as a sequence of sentences in Figure 2-2, $\mathbf{s} = (s_1, \ldots, s_n)$). The model can be viewed as Conditional Random Field (CRF, [12]) in which probability of a labeling is conditioned on the document. Joint labeling of a document and sentences is defined as $\mathbf{y} = (y^d, y^s_1, \ldots, y^s_n)$, where $y^d \in \mathcal{Y}(d)$ is a document label and $y^s_i \in \mathcal{Y}(s), \forall i = 1, \ldots, n$ are labels of individual sentences.[7]

McDonald et al. use structured linear classifier [3] for inference and learning of the

[7]Sets of possible document and sentence labels, $\mathcal{Y}(d)$ and $\mathcal{Y}(s)$, are discrete.

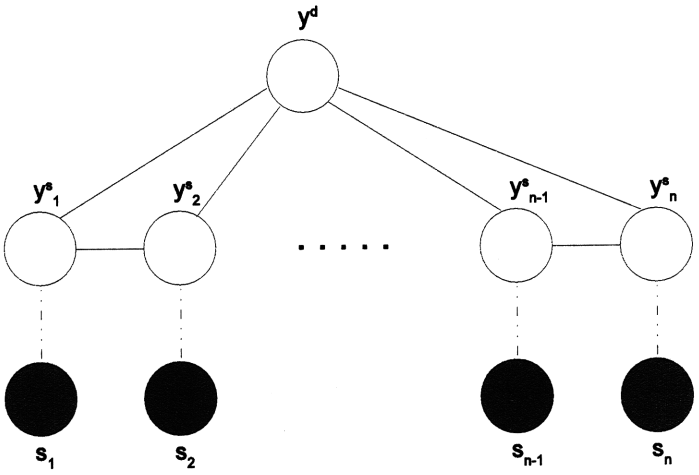

Figure 2-2: Sentence-Document Model for Sentiment Analysis. y^d is document label, while s_i and y_i^s ($\forall i = 1, \ldots, n$) represent i^{th} sentence and sentence label, respectively. Note that sentences are observed.

given model.[8] The score of a labeling $\mathbf{y}$ on a document $\mathbf{s}$ is defined as

$$score(\mathbf{y}, \mathbf{s}) = \sum_{i=2}^{n} score(y^d, y_{i-1}^s, y_i^s, \mathbf{s}), \tag{2.3}$$

where

$$score(y^d, y_{i-1}^s, y_i^s, \mathbf{s}) = \mathbf{w} \cdot \mathbf{f}(y^d, y_{i-1}^s, y_i^s, \mathbf{s}) \tag{2.4}$$

is a score of one clique; $\mathbf{w}$ is a weight vector and $\mathbf{f}(y^d, y_{i-1}^s, y_i^s, \mathbf{s})$ is a feature vector defined on a clique. Authors define a set of binary predicates over the input document $\mathbf{s}$ that includes unigrams, bigrams and trigrams conjoined with their part-of-speech tags. These predicates are combined with label values of a clique to form binary features. Also, they include back-off features to account for the feature sparsity.

Inference problem in this model is to find the labeling $\mathbf{y}$ that gives the highest score on the given document $\mathbf{s}$, i.e.

$$\arg\max_{\mathbf{y}} \; score(\mathbf{y}, \mathbf{s}). \tag{2.5}$$

When document label y^d is fixed, this reduces to a Viterbi algorithm on a chain structure of

[8] Alternatively, CRFs are often used for learning and inference in conditional undirected graphical models.

Input : Training data $\mathcal{T} = (\mathbf{y}_t, \mathbf{s}_t)_{t=1}^{T}$
Init : $\mathbf{w}^{(0)} = 0; i = 0$

Loop : For each iteration $n = 1, \ldots, N$
  Loop : For each training example $t = 1, \ldots, T$
    1. $\mathbf{w}^{(i+1)} = \arg\min_{\mathbf{w}^*} \|\mathbf{w}^* - \mathbf{w}^{(i)}\|$
       s.t. $\forall \mathbf{y}' \in \mathcal{C}$, $score(\mathbf{y}_t, \mathbf{s}_t) - score(\mathbf{y}', \mathbf{s}_t) \geq L(\mathbf{y}_t, \mathbf{y}')$, relative to $\mathbf{w}^*$
    2. $i = i + 1$

Output : $\mathbf{w}^{(N \times T)}$

Figure 2-3: MIRA Learning Algorithm for the Sentence-Document Model

sentence labels. Thus, inference is solved by iterating over possible values for a document label. Model is trained using an inference based large-margin online learning algorithm MIRA [4]. Outline of the algorithm is given in Figure 2-3. In each iteration n, on each training example t, update of the weight vector is performed, such that the new vector is the closest possible vector to the previous one, [9] and the margin between the score of the correct labeling $\mathbf{y}_t$ and the score of a labeling $\mathbf{y}'$ from the contraints set $\mathcal{C}$ is higher than or equal to the loss function $L(\mathbf{y}_t, \mathbf{y}')$. By this training method, the margin between the correct labeling and the labelings that are close to the separating hyperplane (constraint set) is pushed further apart. Loss function is a non-negative function over two sequences that measures how different the sequences are. Authors find that the hamming loss over sentence labels [10] multiplied by the 0-1 loss over document labels works best for this problem.

Finally, McDonald et al. discuss possible extension to a multi-level model, in which inference can be performed similarly, using bottom-up dynamic programming strategy.

2.2.3 Resuts

McDonald et al. test their model on an annotated corpus of 600 reviews from three domains: car seats for children, fitness equipment, and Mp3 players. Reviews are labeled using $\mathcal{Y}(d) = \{pos, neg\}$ set of labels (positive or negative). Sentences are labeled using $\mathcal{Y}(s) = \{pos, neg, neu\}$ set of labels (positive, negative or neutral), where sentences that

[9] in order to preserve influence of previous updates
[10] This is typical for sequence classification problems.

	Car	Fit	Mp3	Total
SENTENCE-CLASSIFIER	54.8	56.8	49.4	53.1
SENTENCE-STRUCTURED	60.5	61.4	55.7	58.8
DOCUMENT→SENTENCE	59.7	61.0	58.3	59.5
JOINT-STRUCTURED	63.5	65.2	60.1	62.6

Table 2.2: Sentence accuracy on the three datasets.

	Car	Fit	Mp3	Total
DOCUMENT-CLASSIFIER	72.8	80.1	87.2	80.3
SENTENCE→DOCUMENT	75.9	80.7	86.1	81.1
JOINT-STRUCTURED	81.5	81.9	85.0	82.8

Table 2.3: Document accuracy on the three datasets.

do not convey sentiment information are labeled as neutral.

They compare the results of the Sentence-Document model (Joint-Structured) with three baselines (Document-Classifier, Sentence-Classifier, Sentence-Structured) and the cascaded models (Sentence→Document, Document→Sentence). **Document-Classifier** model predicts document label only. **Sentence-Classifier** model predicts the label of each sentence independently. **Sentence-Structured** is a sequential model for sentence label prediction. It models dependencies between adjacent sentences. However, it does not depend on the document label. Finally, **Sentence→Document** and **Document→Sentence** models are cascaded models in which the output of the first stage is passed as an input to the second stage.[11] **Sentence-Structured** and **Document-Classifier** models are used for the first stage, as well as in the second stage (in reverse roles), where the feature vectors are augmented with the label information from the first stage.

Tables 2.2 and 2.3 summarize the sentence and document accuracy results of different models on the three datasets, respectively.

From Table 2.2 we can see that modeling dependencies between sentences (**Sentence-Structured**) significantly improves sentence accuracy over the model that assumes their independence (**Sentence-Classifier**). However, the joint model outperforms **Sentence-**

[11] **Sentence→Document** model is similar to the model of Pang and Lee [23].

Structured on all three datasets (statistically significant), showing that document level sentiment information helps sentence accuracy. Authors also suggest a possible scenario when a review contains document level sentiment information, and the goal is to label sentences. They apply the **Joint-Structured** model to classify sentences through a constrained inference in which document label is fixed, and show that the accuracy improves significantly (from 62.6% to 70.3%).

On the other hand, on document level, although **Joint-Structured** model performs better than **Document-Classifier** overall, this result is not consistent across datasets. The improvement is statistically significant only on the *Car* dataset, while the result is worse on *Mp3* dataset. Authors suspect that the cause for such results is overfitting.

Finally, the results of cascaded models are inconsistent and only slightly better than the baselines in both sentence and document accuracy, which shows that information from another level is beneficial. However, this information is passed only once. Passing this information back and forth in both directions, which is the case with the joint model, boosts further the accuracy on both levels.

2.2.4 Comparison to Our Method

Work of McDonald et al. and our work have in common that multi-level analysis is performed in a joint fashion. However, there are some significant differences. Their goal is to infer sentiment on all levels of granularity and improve on each level from mutual sentiment information. On the other hand, our aim is to improve document level sentiment classification using relevance information of substructures. In particular case of sentence-document modeling, the model presented in [19] looks at both document and sentence level sentiment, while we model document sentiment using relevance of sentences. From that perspective, the information from sentence level analysis that is used to improve document level sentiment prediction is slightly different in the two approaches – sentiment in the work of McDonald et al. and relevance in our work. These two are related to some extent. McDonald et al. use $\mathcal{Y} = \{pos, neg, neu\}$ as a set of possible sentence sentiment labels in their experiments, which seems as a further division of the set $\{relevant, irrelevant\}$,

where *irrelevant* is related to *neu* label, and *relevant* is further split into *pos* and *neg* labels. However, although the motivation for relevance stems from the different influence of objective and subjective segments, our model may not necessary learn that division. It will rather learn what best helps the document level sentiment classification.

Another key difference is that McDonald et al. train their model in a fully supervised way, which requires annotation of sentence sentiment, while we model relevance as hidden information. Therefore, it is reasonable that we treat relevance as an auxiliary information to induce document sentiment. McDonald et al. discuss the possibility of modeling sentence sentiment as a hidden variable as a natural extension of their approach, because most reviews contain information about document sentiment, but rarely include sentence sentiment.

2.3 Relevant Regions for Information Extraction

Patwardhan and Riloff [24] describe a two-step method in which relevance regions of the text are detected first, and then extraction patterns are applied on relevant regions.[12] They show an improvement over the standard information extraction approach, in which patterns are applied uniformly on the whole document.

2.3.1 Problem Description and Motivation

The goal of the event-oriented IE system is to extract facts associated with domain-specific events from unstructured text [24]. Two common approaches to building such a system are rule-based and classifier-based approach. The former approach relies on the patterns that can be manually crafted or learned. In order a word or phrase to be extracted, its context (surrounding) should match one of these patterns. The latter approach use machine learning techniques to label the words that should be extracted based on their surrounding words and themselves.

[12] As described in Section 2.3.2, the most reliable extraction patterns are also applied on irrelevant regions in order to make up for the mistakes of the relevance classifier.

While the difference between the two approaches is how the extraction decision is made – hard decision based on pattern matching or soft decision based on feature values (e.g. their weighted combination), the decision is made using only local context in both of them. Patwardhan and Riloff conclude that this decision effectively incorporates two decisions that are performed simultaneously: 1) whether the context is relevant and 2) what should be extracted.

They argue that the decomposition of these two tasks is beneficial. This is due to the ability of deciding relevance based on the broader context. For example, in the following sentences[13]

"John Kerry attacked George Bush."

"Other brave minds that advocated reform had been killed before in that struggle."

the patterns *"<subject> attacked <object>"*, and *"<subject> had been killed"* would likely be applied to extract information on terrorists attacks. However, the "attack" in the first sentence is verbal, and the subject in the second example is not related to a specific physical attack. These wrong extractions of typical patterns might be prevented only if the broader context is looked into. In another example

"the gun was found..."

the fact that the gun was found does not directly imply that it was used in an attack, and most likely it will not be extracted. However, there are contexts in which it should be extracted, and such a decision requires knowing that context.

2.3.2 Method

Patwardhan and Riloff develop a self-trained relevant sentence SVM classifier using a training set of relevant and irrelevant documents from the domain and a few seed patterns. Sentence is chosen as a region size, since it is easy to detect and big enough to contain useful information. Irrelevant documents are used as a pool of irrelevant sentences for training (the assumption is that all sentences in an irrelevant document are irrelevant). Relevant documents are the source of unlabeled sentences. In the initial step, seed patterns (that are

[13] All examples are from [24].

reliable) are applied to relevant documents and sentences that are found to be relevant are used for training. The same amount of irrelevant sentences are drawn from the pool of irrelevant sentences. After SVM classifier is trained, it is applied on relevant documents to produce more relevant sentences that are used for retraining. Again the same amount of irrelevant sentences are drawn and the new SVM is trained. The procedure is repeated for several iterations.

Authors extract patterns based on their semantic affinity to event roles, which measures the tendency of a pattern to extract noun phrases that belong to an event role. The semantic affinity of a pattern p with respect to an event role r_k is defines as [24]:

$$sem_{aff}(p, r_k) = \frac{f(p, r_k)}{\sum_{i=1}^{|R|} f(p, r_i)} log_2 f(p, r_k), \tag{2.6}$$

where R is the set of event roles (including *Other* to account for pattern appearances not related to any role of interest), and $f(p, r_k)$ is the number of appearances of pattern p related to the role r_k.[14] These counts are obtained from the training corpus. For each event role, N patterns with the highest semantic affinity with respect to that role are used for extraction.

To account for the mistakes of the relevance classifier, the most reliable patterns are applied on both relevant and irrelevant sentences. These are called primary patterns. Secondary patterns are applied only on relevant sentences. To distinguish between relevant and irrelevant patterns, the conditional probability of a pattern being relevant is computed based on its appearance in the relevant and irrelevant documents of the training set, and a threshold on this probability is used. Furthermore, another (lower) threshold is used to filter out the least relevant patterns, which are not likely to be useful even if they have high semantic affinity for some roles, because they mainly occur in irrelevant documents.

2.3.3 Resuts

Patwardhan and Riloff evaluate the performance of their IE system on two datasets: the MUC-4 terrorism corpus [31], and a ProMed disease outbreaks corpus.[15] The MUC-4

[14] More details are given in [24].

[15] http://www.promedmail.org

	R1			R2			R3			R4			R5		
	Rec	Pr	F	Rec	Pr	F	Rec	Pr	F	Rec	Pr	F	Rec	Pr	F
ALL	.50	.27	.35	.42	.43	.42	.56	.38	.45	.50	.33	.40	.53	.46	.50
REL	.46	.39	.42	.34	.61	.43	.52	.45	.48	.44	.45	.45	.41	.56	.48
SEL	.48	.39	.43	.36	.58	.45	.56	.46	.50	.46	.44	.45	.43	.53	.48

Table 2.4: MUC-4 Results

dataset consists of 1700 documents (1300 training, 200 development, 200 test). They focus on five roles: *perpetrator individuals, perpetrator organizations, physical targets, victims, and weapons*. Answer keys are used to separate the training set into relevant and irrelevant subsets, i.e. any document containing at least one relevant event is considered relevant. The ProMed dataset contains 125 tuning and 120 test documents for which answer key templates are created. Training set consists of 2000 ProMed documents that are used as relevant, and 4000 more from biomedical abstracts[16] (similar, but different domain) that are used as irrelevant documents. The considered roles are *diseases and victims*.

The authors evaluate their system on extractions themselves, rather than on template creation, which is a complex task and involves other ingredients, such as coreference resolution, that are not focus of their work. They use a head-noun scoring scheme in which an extraction is considered correct if its head noun matches the head noun in the answer key. Pronouns were discarder, since no coreference is involved, and duplicate extractions are counted once.

Some of the results on MUC-4 and ProMed datasets are shown in Table 2.4 and Table 2.5. Columns correspond to precision, recall and F-measure of different roles (listed above). Rows correspond to three variants. In the first variant (*All*), N patterns with highest semantic affinity are applied on all sentences. In the second variant (*Rel*), N patterns are applied only on relevant sentences. Finally, in the third variant (*Sel*), primary patterns are applied on all sentences, while secondary patterns are applied only on relevant sentences. Authors try different number of patterns N (50, 100, 150, and 200). For each role we show only the results for the value of N that gives the best results.

[16] http://www.pubmedcentral.nih.gov

	R1			R2		
	Rec	Pr	F	Rec	Pr	F
ALL	.51	.25	.34	.47	.41	.44
REL	.49	.31	.38	.44	.43	.43
SEL	.50	.29	.36	.46	.41	.44

Table 2.5: ProMed Results

Typically, precision increases and recall decreases when patterns are applied to relevant sentences instead of all sentences. This is expected, since the set of relevant sentences tends to correlate better with the task, while some information, contained in the other sentences, is missed. However, the increase in precision is more significant, leading to the increase in F-measure. Finally, when these two approaches are combined (*Sel* column) by applying only the most reliable patterns to irrelevant sentences, recall increases again (comparing to *Rel*), while precision is the same or slightly worse, leading to an improved F-measure in most cases. Therefore, it is beneficial to distinguish between relevant and irrelevant sentences. Authors also compare this method to AutoSlog-TS IE system [28] and show that the results are similar, although AutoSlog-TS has manually reviewed patterns. In another experiment, when AutoSlog-TS patterns are applied to relevant sentences, there is improvement.

Patwardhan and Riloff also evaluate the performance of the relevant sentence classifier indirectly. Since they do not have relevance annotations of sentences, they induce them from the tuning set by considering a sentence relevant if it contains a string that occurs in the corresponding answer key template, and irrelevant otherwise. Although noisy, [17] this evaluation gives an estimate of the performance, and may also serve to tune the number of iterations of the relevance training procedure. The final accuracy is 82% in the terrorism domain and 63% in the disease outbreak domain. The precision of irrelevant sentences is high in both domains, but the precision of relevant sentences is weak. However, this still leads to an improvement of their IE system over the baseline. Their explanation is that relevance classifier favorably alters the proportion of truly relevant and irrelevant sentences

[17] For example, multiple sentences may contain answer, but only some of them might be relevant. Also, some sentences, primarily those containing pronouns coreferenced with the answer, may be relevant, but not regarded as such for this evaluation.

among those labeled as relevant. For example, the fraction of relevant sentences increases from 17% to 46% in the terrorism domain, and from 28% to 41% in the disease outbreak domain. Also, answer keys often contain multiple acceptable answers, which increases the chance of finding one within sentences labeled as relevant.

2.3.4 Comparison to Our Method and Discussion

Patwardhan and Riloff advocate the benefit of the decoupled model in which the relevance of the broader region helps to improve extraction accuracy of phrase. However, they compare it against models that simultaneously perform relevance identification and extraction locally (looking only at the phrase, i.e. local context of a word). A true joint model that corresponds to their decoupled approach would simultaneously try to decide whether a region is relevant and extract phrases form it. We suggest such approach as a potential future work (Chapter 5). Another benefit from decoupling that they point to is the simplification of the learning process, since models at each stage are simpler.

Their approach is similar to the approach of Pang and Lee [23] in that they both develop a two-step method in which the first step is to classify document segments into relevant and irrelevant for the second step (main task), and apply the second step on the relevant parts of the document. Main task differs in these two works – it is document polarity classification in the work of Pang and Lee and information extraction in the work of Patwardhan and Riloff. Another difference is that Patwardhan and Riloff train the relevance model in a semi-supervised fashion, using only a handful of seed patterns and relevance of documents, thus not requiring relevance annotation – which is also a property of our approach.

Chapter 3

Relevance-Aware Classification

In this chapter, we introduce the relevance-aware classification method. We define the relevance-aware model in Section 3.1, and describe its implementation as a Conditional Bayesian Network (CBN) with hidden variables in Section 3.2. Then, we discuss methods for combining features of relevant segments in Section 3.3. Finally, we describe the implementation of the relevance-aware model as a Conditional Random Field (CRF) with hidden variables in Section 3.4.

3.1 Relevance-Aware Model

The structure of our model is shown in Figure 3-1. Our goal is to learn a mapping from documents $\mathbf{x}$ to labels y, where y belongs to a set $Y = \{1, \ldots, m\}$. For instance, in the case of sentiment classification $Y = \{positive, negative\}$. We model a document as a sequence of segments $\mathbf{x} = (x_1, \ldots . x_n)$. The optimal granularity of a segment is further explored in Section 4.3; for concreteness, assume that a segment corresponds to a paragraph.

Each segment is associated with two feature vectors: $f(x_i)$ is used for label prediction and $g(x_i)$ is used for relevance prediction. The training set consists of N document-label pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$, where $y_1, \ldots, y_N \in Y$, and each document $\mathbf{x}_k$ is split into segments $\mathbf{x}_k = (x_{k,1}, \ldots, x_{k,n_k})$. In addition, for each segment i we introduce a latent binary random variable $z_i \in \{0, 1\}$, which captures its relevance; a value of $z_i = 0$ indicates that a segment is not relevant for classification, and $z_i = 1$ indicates that it is. The document

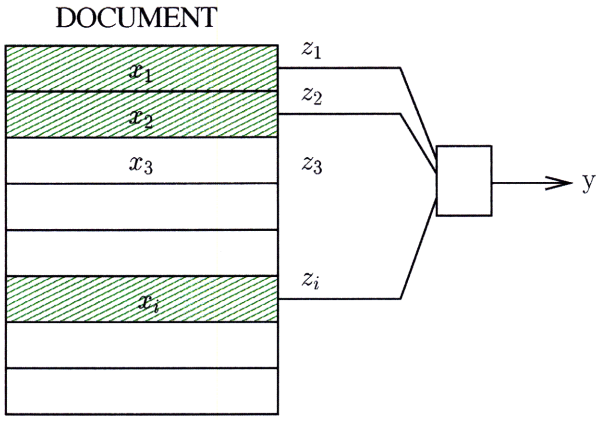

Figure 3-1: The Structure of a Relevance-Aware Text Classifier: y is a classification label, while x_i and z_i represent a segment and its relevance, respectively.

relevance sequence $z_1 \ldots z_n$ is denoted by $\mathbf{z}$.

Given these definitions, we define a probabilistic model parameterized over weight vectors $\mathbf{w_y}$ and $\mathbf{w_r}$, and scalars θ and θ_0. The model is a probabilistic formulation of the following two step procedure:

- **Relevance prediction**: Use the document $\mathbf{x}_i$ to decide on the relevance or irrelevance of the segments in the text (i.e., predict a value for the relevance sequence $\mathbf{z}$)

- **Label prediction**: Given a relevance sequence, use only the relevant segments in this sequence to predict the label of the document.

3.2 Conditional Bayesian Network Implementation

In this section, we define a relevance-aware classification model implemented as a Conditional Bayesian Network (CBN) with hidden variables. We first introduce the components of the model and describe its parametrization. Next, we explain how gradient descent and sampling are employed for parameter estimation.

A natural way of formulating the above two steps is via two conditional distributions, one for $p(\mathbf{z}|\mathbf{x})$ and one of $p(y|\mathbf{z}, \mathbf{x})$. As is standard in probabilistic classification, we use a logistic regression model for each step [1, 27]. In what follows we describe these two

distributions, and how to combine them into one global model.

Relevance prediction: It is reasonable to assume that consecutive segments will tend to have the same relevance value. Thus, in modeling relevance, we employ a Markov chain assumption, such that z_i may depend on the value of z_{i-1}. The distribution $p(\mathbf{z}|\mathbf{x})$ is then modeled as:[1]

$$P(\mathbf{z}|\mathbf{x}; \mathbf{w_r}, \bar{\theta}) = \prod_{i=1}^{n} P(z_i|z_{i-1}, \mathbf{x}; \mathbf{w_r}, \bar{\theta}) \,, \tag{3.1}$$

where each conditional distribution above is modeled via logistic-regression

$$P(z_i|z_{i-1}, \mathbf{x}; \mathbf{w_r}, \bar{\theta}) = \frac{e^{z_i \xi(x_i, \mathbf{w_r}, \bar{\theta})}}{1 + e^{\xi(x_i, \mathbf{w_r}, \bar{\theta})}} \tag{3.2}$$

and

$$\xi(x_i, \mathbf{w_r}, \bar{\theta}) = \begin{cases} \mathbf{w_r} \cdot g(x_i) + \theta z_{i-1} & , i > 1 \\ \mathbf{w_r} \cdot g(x_i) + \theta_0 & , i = 1 \end{cases}$$

Here, $\mathbf{w_r}$ is a weight vector, θ is a constant that ties the relevance of a current segment with the relevance of the previous segment, and θ_0 is a relevance bias for the first segment.

Label prediction: Given the values of the relevance sequence, the label is decided based only on the relevant segments. Here again it is natural to work with a logistic regression model defined by

$$P(y|\mathbf{z}, \mathbf{x}; \mathbf{w_y}) = \frac{e^{\mathbf{w_y} \cdot \sum_{i=1}^{n} z_i f(x_i)}}{\sum_{y'=1}^{m} e^{\mathbf{w_{y'}} \cdot \sum_{i=1}^{n} z_i f(x_i)}} \,, \tag{3.3}$$

where for each $y \in Y$, $\mathbf{w_y}$ is a parameter vector.[2] Note that in the above sum, irrelevant segments will be multiplied by zero (z_i), and hence will not affect the decision regarding the label.

A Global Model: The two distributions defined above imply the following joint distribution over $\mathbf{z}$ and y

$$p(y, \mathbf{z}|\mathbf{x}) = p(y|\mathbf{z}, \mathbf{x}) p(\mathbf{z}|\mathbf{x}) \tag{3.4}$$

where we have left out the parameters for brevity. It is easy to see that the above model for

[1] $\bar{\theta}$ is a notation that unifies parameters θ and θ_0.

[2] Note that there is a parameter vector for each possible label.

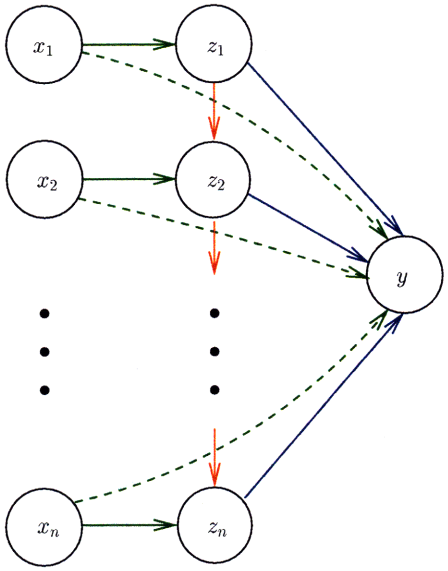

Figure 3-2: Conditional Bayesian Network for Relevance Classification

y and $\mathbf{z}$ constitutes a Bayesian network, as illustrated in Figure 3-2. Marginalization of this distribution yields the probability of the label y given the document

$$p(y|\mathbf{x}) = \sum_{\mathbf{z}} p(y|\mathbf{z}, \mathbf{x}) p(\mathbf{z}|\mathbf{x}) \tag{3.5}$$

The model formulation presented above allows us to learn relevance and label predictions jointly. Since the goal is to minimize the label prediction error, the parametrization of the relevance component is driven by the performance of the target classification.

3.2.1 Parameter Estimation

The objective function that we aim to minimize is

$$\bar{L}(\mathbf{w_r}, \mathbf{w_y}, \bar{\theta}) = -\sum_{k=1}^{N} \log P(y_k|\mathbf{x}_k; \mathbf{w_r}, \mathbf{w_y}, \bar{\theta}) + \frac{C_r}{2}(\| \mathbf{w_r}^2 \| + \theta^2 + \theta_0^2) + \frac{C_y}{2} \| \mathbf{w_y}^2 \| \tag{3.6}$$

The first term in (3.6) is the log-likelihood of the data. The second and third terms are regularization terms.

We use stochastic gradient descent method to train the model. In this framework, one training example $(\mathbf{x}_k, y_k)$ is observed at a time. Therefore, we refine the objective function at each step to be

$$L(\mathbf{w_r}, \mathbf{w_y}, \bar{\theta}) = -\log P(y_k|\mathbf{x}_k; \mathbf{w_r}, \mathbf{w_y}, \bar{\theta}) + \frac{C_r}{2N}(\| \mathbf{w_r}^2 \| + \theta^2 + \theta_0^2) + \frac{C_y}{2N} \| \mathbf{w_y}^2 \| \tag{3.7}$$

During each training iteration we perform updates on all the training examples. We repeat this process, changing the order of examples from one iteration to another.

To perform gradient descent method, gradients with respect to $\nu \in \{\mathbf{w_y}, \mathbf{w_r}, \theta, \theta_0\}$ are required.

$$\frac{\partial L}{\partial \nu} = -\frac{1}{P(y_k|\mathbf{x}_k)} \frac{\partial P(y_k|\mathbf{x}_k)}{\partial \nu} + \frac{C}{N}\nu \tag{3.8}$$

where $C = C_y$ for $\nu = \mathbf{w_y}$, and $C = C_r$ otherwise. In Figure 3-3, we specify partial derivatives $\frac{\partial P(y_k|\mathbf{x}_k)}{\partial \nu}$ for each ν.

3.2.2 Sampling

Efficient computation of gradients as specified in Eq. 3.8 is challenging. The term $\frac{1}{P(y_k|\mathbf{x}_k)} \frac{\partial P(y_k|\mathbf{x}_k)}{\partial \nu}$ contains summation over all possible relevance sequences (3.9, 3.10). Enumerating all sequences takes exponential time with respect to the number of segments. On an arbitrary long text the exact approach is not feasible. Therefore, we approximate this term using Gibbs sampling method [2]. For $\nu = \mathbf{w_y}$, it can be rewritten as (using Eq. 3.9)

$$\begin{aligned} \frac{1}{P(y_k|\mathbf{x}_k)} \frac{\partial P(y_k|\mathbf{x}_k)}{\partial \mathbf{w_y}} &= \sum_{\mathbf{z}} \left[\frac{P(\mathbf{z}|\mathbf{x}_k) P(y_k|\mathbf{z}, \mathbf{x}_k)}{P(y_k|\mathbf{x}_k)} \cdot (\delta_{y,y_k} - P(y|\mathbf{z}, \mathbf{x}_k)) \sum_{i=1}^{n_k} z_i f(x_{k,i}) \right] \\ &= \sum_{\mathbf{z}} P(\mathbf{z}|y_k, \mathbf{x}_k) \left[(\delta_{y,y_k} - P(y|\mathbf{z}, \mathbf{x}_k)) \cdot \sum_{i=1}^{n_k} z_i f(x_{k,i}) \right] \end{aligned} \tag{3.14}$$

$$\frac{\partial P(y_k|\mathbf{x}_k)}{\partial \mathbf{w_y}} = \sum_{\mathbf{z}} \left[P(\mathbf{z}|\mathbf{x}_k) P(y_k|\mathbf{z}, \mathbf{x}_k) \left(\delta_{y,y_k} - P(y|\mathbf{z}, \mathbf{x}_k)\right) \sum_{i=1}^{n_k} z_i f(x_{k,i}) \right], \delta_{y,y_k} = \begin{cases} 1 & , y = y_k \\ 0 & o.w. \end{cases} \tag{3.9}$$

$$\frac{\partial P(y_k|\mathbf{x}_k)}{\partial \mu} = \sum_{\mathbf{z}} P(\mathbf{z}|\mathbf{x}_k) \left[\sum_{i=1}^{n_k} \frac{\frac{\partial P(z_i|z_{i-1}, \mathbf{x}_k)}{\partial \mu}}{P(z_i|z_{i-1}, \mathbf{x}_k)} \right] P(y_k|\mathbf{z}, \mathbf{x}_k) \qquad , \mu \in \{\mathbf{w_r}, \theta, \theta_0\} \tag{3.10}$$

$$\frac{\partial P(y_k|\mathbf{x}_k)}{\partial \mathbf{w_r}} = \sum_{\mathbf{z}} P(\mathbf{z}|\mathbf{x}_k) \left[\sum_{i=1}^{n_k} g(x_{k,i})(-1)^{z_i} \left(P(z_i|z_{i-1}, \mathbf{x}_k) - 1\right) \right] P(y_k|\mathbf{z}, \mathbf{x}_k) \tag{3.11}$$

$$\frac{\partial P(y_k|\mathbf{x}_k)}{\partial \theta} = \sum_{\mathbf{z}} P(\mathbf{z}|\mathbf{x}_k) \left[\sum_{i=2}^{n_k} z_{i-1}(-1)^{z_i} \left(P(z_i|z_{i-1}, \mathbf{x}_k) - 1\right) \right] P(y_k|\mathbf{z}, \mathbf{x}_k) \tag{3.12}$$

$$\frac{\partial P(y_k|\mathbf{x}_k)}{\partial \theta_0} = \sum_{\mathbf{z}} P(\mathbf{z}|\mathbf{x}_k)(-1)^{z_1} \left(P(z_1|\mathbf{x}_k) - 1\right) P(y_k|\mathbf{z}, \mathbf{x}_k) \tag{3.13}$$

Figure 3-3: Derivatives of $P(y_k|\mathbf{x}_k)$ with respect to $\nu \in \{\mathbf{w_y}, \mathbf{w_r}, \theta, \theta_0\}$

For $\mu \in \{\mathbf{w_r}, \theta, \theta_0\}$, we can write (using Eq. 3.10)

$$\begin{aligned} \frac{1}{P(y_k|\mathbf{x}_k)} \frac{\partial P(y_k|\mathbf{x}_k)}{\partial \mu} &= \sum_{\mathbf{z}} \frac{P(\mathbf{z}|\mathbf{x}_k) P(y_k|\mathbf{z}, \mathbf{x}_k)}{P(y_k|\mathbf{x}_k)} \left[\sum_{i=1}^{n_k} \frac{\frac{\partial P(z_i|z_{i-1}, \mathbf{x}_k)}{\partial \mu}}{P(z_i|z_{i-1}, \mathbf{x}_k)} \right] \\ &= \sum_{\mathbf{z}} P(\mathbf{z}|y_k, \mathbf{x}_k) \left[\sum_{i=1}^{n_k} \frac{\frac{\partial P(z_i|z_{i-1}, \mathbf{x}_k)}{\partial \mu}}{P(z_i|z_{i-1}, \mathbf{x}_k)} \right] \end{aligned} \tag{3.15}$$

In both (3.14) and (3.15), $P(\mathbf{z}|y_k, \mathbf{x}_k)$ is a distribution from which we sample sequence $\mathbf{z}$. Each z_i is drawn independently from a distribution $P(z_i|\mathbf{z}_{-i}, y_k, \mathbf{x}_k)$, where $\mathbf{z}_{-i}$ represents relevance sequence $\mathbf{z}$ excluding z_i. This distribution is computed by calculating $P(z_i = 1, \mathbf{z}_{-i}, y_k|\mathbf{x}_k)$ and $P(z_i = 0, \mathbf{z}_{-i}, y_k|\mathbf{x}_k)$ first, and normalizing them. Expressions

in square brackets in (3.14) and (3.15) are calculated using sampled $\mathbf{z}$ values, and then averaged to obtain an approximation of the sum in Eq. 3.14 or 3.15.

Gibbs sampling is also applied to approximate the summation on the right side of Equation 3.5 in order to compute $p(y|\mathbf{x})$ during inference. Sequence $\mathbf{z}$ is sampled from the distribution $p(\mathbf{z}|\mathbf{x})$, and $p(y|\mathbf{z},\mathbf{x})$ is calculated for each sample, and then averaged to obtain an approximated value of $p(y|\mathbf{x})$. Similarly as above, to obtain sample $\mathbf{z}$, each z_i is drawn independently from a distribution $P(z_i|\mathbf{z}_{-i},\mathbf{x}_k)$, which is computed by calculating $P(z_i=1,\mathbf{z}_{-i}|\mathbf{x}_k)$ and $P(z_i=0,\mathbf{z}_{-i}|\mathbf{x}_k)$ first, and then normalizing them.

To obtain a sample $\mathbf{z}$, we draw z_i values for $i=1,\ldots,n_k$ in that order. Initial values are set randomly. After the first sample is obtained, every next sample is drawn using previous one as initial value. Finally, initial r samples are discarded, since they will not come from the desired distribution.

3.2.3 Parameter Tuning

Since the objective function is not convex, and the search space contains many local minima, the gradient descent method may find a suboptimal solution. We address this issue, by performing several random restarts and randomized order of training examples.

We tune regularization constants C_r and C_y on the development set (see Chapter 4 for more details on the training procedure and tuning).

3.3 Integrating Features of Relevant Segments

The expression $\sum_{i=1}^{n} z_i f(x_i)$ creates a document feature vector for each relevance sequence $\mathbf{z}$ by summing feature vectors $f(x_i)$ of relevant segments ($z_i=1$). While $f(x_i)$ is a binary feature vector, the sum will not have the same property. Even further, the magnitude of features will be proportional to the number of relevant segments. This is undesired, because the feature vectors created for different relevance sequences (and also for different documents) will not be on the same scale.

Here, we present several ways of combining segment features into one feature vector $f(\mathbf{x},\mathbf{z})$:

Default Features of relevant segments are simply summed as above:

$$f(\mathbf{x}, \mathbf{z}) = \sum_{i=1}^{n} z_i f(x_i) \tag{3.16}$$

1-Normalization After the summation, the feature vector is normalized by its L_1 norm:

$$f_{1N}(\mathbf{x}, \mathbf{z}) = \frac{f(\mathbf{x}, \mathbf{z})}{\|f(\mathbf{x}, \mathbf{z})\|_1} \tag{3.17}$$

2-Normalization After the summation, the feature vector is normalized by its L_2 norm:

$$f_{2N}(\mathbf{x}, \mathbf{z}) = \frac{f(\mathbf{x}, \mathbf{z})}{\|f(\mathbf{x}, \mathbf{z})\|_2} \tag{3.18}$$

Clipping After the summation, the feature vector is binarized, by clipping all values greater than 1 to the value 1. The resulting feature is then 1 whenever it appears at least once in the relevant segments. We can write

$$f_C(\mathbf{x}, \mathbf{z})_i = \begin{cases} 1, & f(\mathbf{x}, \mathbf{z})_i > 0 \\ 0, & f(\mathbf{x}, \mathbf{z})_i = 0 \end{cases} \tag{3.19}$$

for each feature i, or alternatively

$$f_C(\mathbf{x}, \mathbf{z}) = \bigvee_{i=1}^{n} z_i f(x_i). \tag{3.20}$$

Averaging Each feature has the value that is an average of its values across relevant segments, i.e., the feature vector is divided by the number of relevant segments after the summation:

$$f_A(\mathbf{x}, \mathbf{z}) = \frac{1}{\sum_{i=1}^{n} z_i} f(\mathbf{x}, \mathbf{z}) = \frac{1}{\sum_{i=1}^{n} z_i} \sum_{i=1}^{n} z_i f(x_i) \tag{3.21}$$

For the convenience, we keep the notation $\sum_{i=1}^{n} z_i f(x_i)$ in all equations, but note that it has different interpretation depending on the method for combining features that is actually

employed. We evaluate these techniques in our experiments (Chapter 4.1). The details of optimization in all cases are nearly identical to those in Section 3.2.1.

We can apply any of these methods in the approximate inference and training using the sampling method, since we can directly compute feature vector for each relevance sequence sample by applying the appropriate formula above. While the approximate inference is a valid approach for both CBN and CRF implementations, the advantage of CRF is that belief propagation algorithm can be applied for exact inference if the factors in the model are small. However, this is not the case when **normalization** and **clipping** options are used. The reason is that, in these two variants, the feature vector expression cannot be decomposed as a product of terms pertaining to each segment. In other words, one big factor that contains all segments is created. Although CRF with **averaging** option seems to have the same issue, path aggregation technique [22] can be applied to model the normalization factor $\sum_{i=1}^{n} z_i$ in an efficient way. Therefore, we implement this method (Section 3.4.2). On the other hand, in the CBN implementation, the factors describing probability distributions $p(\mathbf{z}|\mathbf{x})$ and and $p(y|\mathbf{z},\mathbf{x})$ are locally normalized, creating one big factor of all segments. Therefore, we need to employ approximate inference, regardless of the method for combining features.

3.4 Conditional Random Field Implementation

In this section, we describe the implementation of the relevance-aware model (Figure 3-1) as a Conditional Random Field (CRF) with latent variables [26]. This was suggested in McDonald et al. [19] as an extension of their approach to learning with hidden variables. Graphical representation of this model is the same as in Figure 2-2,[3] which is essentially an undirected version of the model in Figure 3-2. [4] However, McDonald et al. are interested in both recovering document and sentence level labels in the scenario of partially labeled data, while we are primarily focused on fully unsupervised segment relevance modeling for document classification.

[3] Segments of any level can stand for sentences in Fig. 2-2 .

[4] Links between segments and document label, that exist in Fig. 3-2, are not shown in Fig. 2-2, but they are inevitably present, since the segments are conditioned on and therefore present in all cliques.

Under the CRF framework, we define the joint distribution over $\mathbf{z}$ and y directly as a log-linear [5] model: [6]

$$p(y, \mathbf{z}|\mathbf{x}) = \frac{1}{Z}\left[\prod_{i=1}^{n} e^{z_i \xi(x_i, \mathbf{w_r}, \bar{\theta})}\right] e^{\mathbf{w_y} \cdot \sum_{i=1}^{n} z_i f(x_i)}, \tag{3.22}$$

where

$$Z = \sum_{y'=1}^{m} \sum_{\mathbf{z}} \left[\prod_{i=1}^{n} e^{z_i \xi(x_i, \mathbf{w_r}, \bar{\theta})}\right] e^{\mathbf{w_{y'}} \cdot \sum_{i=1}^{n} z_i f(x_i)} \tag{3.23}$$

is a normalization factor, i.e., the partition function. Note that $e^{z_i \xi(x_i, \mathbf{w_r}, \bar{\theta})}$ and $e^{\mathbf{w_y} \cdot \sum_{i=1}^{n} z_i f(x_i)}$ correspond to the numerators of the Equations 3.2 and 3.3, respectively, but the normalization is global. Therefore, distributions $P(z_i|z_{i-1}, \mathbf{x})$, $P(\mathbf{z}|\mathbf{x})$ and $P(y|\mathbf{z}, \mathbf{x})$ are not modeled here directly, and they cannot be computed using Equations 3.2, 3.1 and 3.3 (there is no justification for that). Finally, marginalization of the joint distribution $p(y, \mathbf{z}|\mathbf{x})$ yields

$$p(y|\mathbf{x}) = \sum_{\mathbf{z}} p(y, \mathbf{z}|\mathbf{x}) = \frac{1}{Z} \sum_{\mathbf{z}} \left[\prod_{i=1}^{n} e^{z_i \xi(x_i, \mathbf{w_r}, \bar{\theta})}\right] e^{\mathbf{w_y} \cdot \sum_{i=1}^{n} z_i f(x_i)}. \tag{3.24}$$

For the later convenience, we define function φ as

$$\varphi(y|\mathbf{x}) = \sum_{\mathbf{z}} \left[\prod_{i=1}^{n} e^{z_i \xi(x_i, \mathbf{w_r}, \bar{\theta})}\right] e^{\mathbf{w_y} \cdot \sum_{i=1}^{n} z_i f(x_i)}, \tag{3.25}$$

and express $p(y|\mathbf{x})$ as

$$p(y|\mathbf{x}) = \frac{\varphi(y|\mathbf{x})}{\sum_{y'=1}^{m} \varphi(y'|\mathbf{x})}, \tag{3.26}$$

and Z as

$$Z = \sum_{y'=1}^{m} \varphi(y'|\mathbf{x}). \tag{3.27}$$

[5] $p(y, \mathbf{z}|\mathbf{x}) = \frac{e^{\psi(y, \mathbf{z}, \mathbf{x}, \mathbf{w_y}, \mathbf{w_r}, \theta, \theta_0)}}{\sum_{y'=1}^{m} \sum_{\mathbf{z}} e^{\psi(y', \mathbf{z}, \mathbf{x}, \mathbf{w_{y'}}, \mathbf{w_r}, \theta, \theta_0)}}$, where
$\psi(y, \mathbf{z}, \mathbf{x}, \mathbf{w_y}, \mathbf{w_r}, \theta, \theta_0) = \mathbf{w_r} \cdot \sum_{i=1}^{n} z_i g(x_i) + \theta \cdot \sum_{i=2}^{n} z_i z_{i-1} + \theta_0 \cdot z_i + \mathbf{w_y} \cdot \sum_{i=1}^{n} z_i f(x_i)$ is a potential function linear in the parameters $\mathbf{w_y}$, $\mathbf{w_r}$, θ, and θ_0.

[6] Parameters are left out from probability expressions for brevity.

We also introduce the following definitions, analogous to Eq. 3.2, 3.1, and 3.3:

$$F(z_i|z_{i-1}, \mathbf{x}; \mathbf{w_r}, \bar{\theta}) = e^{z_i \xi(x_i, \mathbf{w_r}, \bar{\theta})} \tag{3.28}$$

$$F(\mathbf{z}|\mathbf{x}; \mathbf{w_r}, \bar{\theta}) = \prod_{i=1}^{n} F(z_i|z_{i-1}, \mathbf{x}; \mathbf{w_r}) \tag{3.29}$$

$$F(y|\mathbf{z}, \mathbf{x}; \mathbf{w_y}) = e^{\mathbf{w_y} \cdot \sum_{i=1}^{n} z_i f(x_i)}, \tag{3.30}$$

and write

$$\varphi(y|\mathbf{x}) = \sum_{\mathbf{z}} F(\mathbf{z}|\mathbf{x}; \mathbf{w_r}, \bar{\theta}) F(y|\mathbf{z}, \mathbf{x}; \mathbf{w_y}) \tag{3.31}$$

analogous to Eq. 3.5.

Although maximal cliques in Figure 2-2 contain three nodes (document label and labels of two neighboring sentences), unlike McDonald et al., who make no further assumptions and have features combining all three labels, we break the distribution over a clique into three pairwise factors – one that relate relevance labels of adjacent sentences (z_i and z_{i-1}), and two that relate the document label with the relevance label of each sentence (y and z_i, y and z_{i-1}). In other words, we put additional constraint on the form of the distribution, such that it has the same properties as the distribution modeled with the Conditional Bayesian Network. Therefore, graphical model in Figure 2-2 is not the most natural representation of the process described by our relevance-aware model, justifying the CBN implementation.

3.4.1 Parameter Estimation

Similar as in Section 3.2.1, we use stochastic gradient descent method to minimize the objective function given in Eq. 3.7. Gradients of the objective function with respect to $\nu \in \{\mathbf{w_y}, \mathbf{w_r}, \theta, \theta_0\}$ can be written as

$$\frac{\partial L}{\partial \nu} = -\frac{\partial \, log P(y_k|\mathbf{x}_k)}{\partial \nu} + \frac{C}{N}\nu, \tag{3.32}$$

$$\frac{\partial \varphi(y_k|\mathbf{x}_k)}{\partial \mathbf{w_y}} = \sum_{\mathbf{z}} F(\mathbf{z}|\mathbf{x}_k) \left[\sum_{i=1}^{n_k} z_i f(x_{k,i}) \right] F(y_k|\mathbf{z}, \mathbf{x}_k), \qquad y = y_k \tag{3.35}$$

$$\frac{\partial \varphi(y_k|\mathbf{x}_k)}{\partial \mathbf{w_y}} = 0, \qquad y \neq y_k$$

$$\frac{\partial \varphi(y_k|\mathbf{x}_k)}{\partial \mu} = \sum_{\mathbf{z}} F(\mathbf{z}|\mathbf{x}_k) \left[\sum_{i=1}^{n_k} \frac{\frac{\partial F(z_i|z_{i-1},\mathbf{x}_k)}{\partial \mu}}{F(z_i|z_{i-1}, \mathbf{x}_k)} \right] F(y_k|\mathbf{z}, \mathbf{x}_k) \qquad , \mu \in \{\mathbf{w_r}, \theta, \theta_0\} \tag{3.36}$$

$$\frac{\partial \varphi(y_k|\mathbf{x}_k)}{\partial \mathbf{w_r}} = \sum_{\mathbf{z}} F(\mathbf{z}|\mathbf{x}_k) \left[\sum_{i=1}^{n_k} z_i g(x_{k,i}) \right] F(y_k|\mathbf{z}, \mathbf{x}_k) \tag{3.37}$$

$$\frac{\partial \varphi(y_k|\mathbf{x}_k)}{\partial \theta} = \sum_{\mathbf{z}} F(\mathbf{z}|\mathbf{x}_k) \left[\sum_{i=2}^{n_k} z_i z_{i-1} \right] F(y_k|\mathbf{z}, \mathbf{x}_k) \tag{3.38}$$

$$\frac{\partial \varphi(y_k|\mathbf{x}_k)}{\partial \theta_0} = \sum_{\mathbf{z}} F(\mathbf{z}|\mathbf{x}_k)\ z_1\ F(y_k|\mathbf{z}, \mathbf{x}_k) \tag{3.39}$$

Figure 3-4: Derivatives of $\varphi(y_k|\mathbf{x}_k)$ with respect to $\nu \in \{\mathbf{w_y}, \mathbf{w_r}, \theta, \theta_0\}$

where $C = C_y$ for $\nu = \mathbf{w_y}$ and $C = C_r$, otherwise. From Eq. 3.26, we can write

$$logP(y_k|\mathbf{x}_k) = log\, \varphi(y_k|\mathbf{x}_k) - log \sum_{y'=1}^{m} \varphi(y'|\mathbf{x}_k) \tag{3.33}$$

and

$$\frac{\partial\, logP(y_k|\mathbf{x}_k)}{\partial \nu} = \frac{1}{\varphi(y_k|\mathbf{x}_k)} \frac{\partial \varphi(y_k|\mathbf{x}_k)}{\partial \nu} - \frac{1}{\sum_{y'=1}^{m} \varphi(y'|\mathbf{x}_k)} \sum_{y'=1}^{m} \frac{\partial \varphi(y'|\mathbf{x}_k)}{\partial \nu}. \tag{3.34}$$

In Figure 3-4, we specify partial derivatives $\frac{\partial \varphi(y_k|\mathbf{x}_k)}{\partial \nu}$ for each ν.

3.4.2 Path Aggregation

With the **averaging** method for combining features of relevant segments, the function $\varphi(y|\mathbf{x})$ looks like

$$\varphi(y|\mathbf{x}) = \sum_{\mathbf{z}} \left[\prod_{i=1}^{n} e^{z_i \xi(x_i, \mathbf{w_r}, \bar{\theta})} \right] e^{\frac{1}{\sum_{i=1}^{n} z_i} \mathbf{w_y} \cdot \sum_{i=1}^{n} z_i f(x_i)}. \tag{3.40}$$

The normalization factor $\sum_{i=1}^{n} z_i$, although seemingly creating a huge factor over all segments, can be handled using path aggregation [22]. We introduce a "counting" variable r_k for each segment k as

$$r_k = \sum_{i=1}^{k} z_i \tag{3.41}$$

Variable r_k summarizes the information of the relevance sequence up to the position k. We can also express it in the following way:

$$r_k = r_{k-1} + z_k, \tag{3.42}$$

where $r_0 = 0$ is introduced for this equation to hold for any $k = 1, \ldots, n$. Obviously variable r_k at the position k only depends on the variable r_{k-1} at the previous position and the relevance z_k at that position. [7] Now, we can write:

$$\varphi(y|\mathbf{x}) = \sum_{\mathbf{z}} \left[\prod_{i=1}^{n} e^{z_i \xi(x_i, \mathbf{w_r}, \bar{\theta})} \right] e^{\frac{1}{r_n} \mathbf{w_y} \cdot \sum_{i=1}^{n} z_i f(x_i)}, \tag{3.43}$$

and see that the function $F(y|\mathbf{z}, \mathbf{x}; \mathbf{w_y})$ can break into the product of factors that depend on one segment and the variable r_n:

$$F(y|\mathbf{z}, \mathbf{x}; \mathbf{w_y}) = e^{\frac{1}{r_n} \mathbf{w_y} \cdot \sum_{i=1}^{n} z_i f(x_i)} = \prod_{i=1}^{n} e^{\frac{1}{r_n} \mathbf{w_y} z_i f(x_i)}, \tag{3.44}$$

which enables efficient inference.

Graphical representation of our CRF model with path aggregation is shown in Figure 3-

[7] This creates factors of size three.

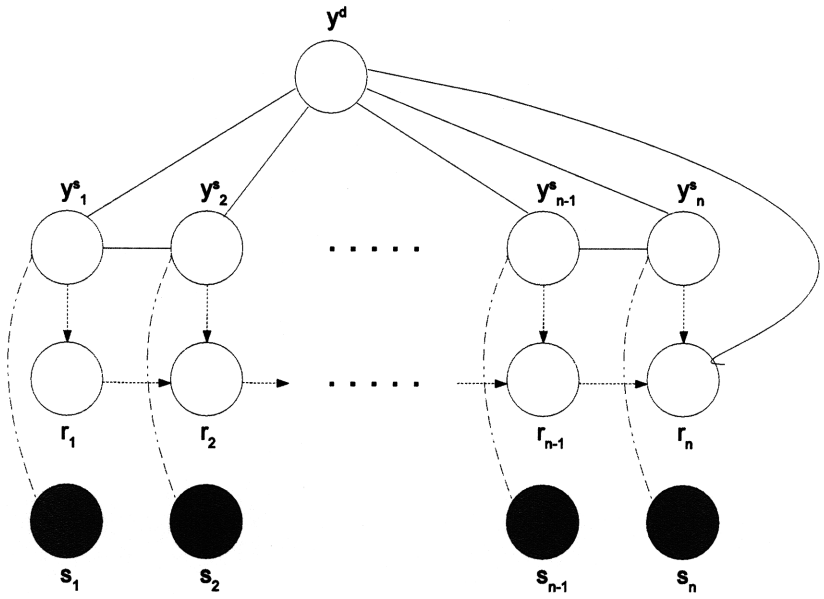

Figure 3-5: CRF Model with Path Aggregation

5. Deterministic dependencies are shown with dashed lines, and the edges are directed towards the variables that are deterministically computed based on the values of other variables. We can see that the label y^d depends on the variable r_n, rather than on the whole relevance sequence.

3.4.3 Efficient Inference and Training

In the case of **averaging**, function $\varphi(y|\mathbf{x})$ (Equation 3.43) can be computed efficiently using belief propagation algorithm, leading to an efficient inference. We develop a dynamic programming solution specific to this problem, that directly computes $\varphi(y|\mathbf{x})$.

Note that $\varphi(y|\mathbf{x})$ can be written as

$$\begin{aligned}\varphi(y|\mathbf{x}) &= \sum_{\mathbf{z}} \prod_{i=1}^{n} e^{z_i\left[\xi(x_i,\mathbf{w_r},\bar{\theta})+\frac{1}{r_n}\mathbf{w_y}\cdot\sum_{i=1}^{n} z_i f(x_i)\right]} \\ &= \sum_{r_n=0}^{n} \sum_{\mathbf{z},\sum_{i=1}^{n} z_i=r_n} \prod_{i=1}^{n} e^{z_i\left[\xi(x_i,\mathbf{w_r},\bar{\theta})+\frac{1}{r_n}\mathbf{w_y}\cdot\sum_{i=1}^{n} z_i f(x_i)\right]} \end{aligned} \tag{3.45}$$

Let S be a table of $n \times (n+1) \times (n+1)$ values, such that

$$S(k, r_k, r_n) = \sum_{z_1,\ldots z_k, \sum_{i=1}^{k} z_i = r_k} \prod_{i=1}^{k} e^{z_i \left[\xi(x_i, \mathbf{w_r}, \bar{\theta}) + \frac{1}{r_n} \mathbf{w_y} \cdot \sum_{i=1}^{n} z_i f(x_i)\right]} \tag{3.46}$$

is a partial sum, where the summation is over all length-k relevance sub-paths (first k segments) with exactly r_k relevant segments and normalization factor r_n. Here, $k \in \{1, \ldots, n\}$, $r_k \in \{0, \ldots, n\}$, and $r_n \in \{0, \ldots, n\}$. Therefore,

$$\varphi(y|\mathbf{x}) = \sum_{r_n=0}^{n} S(n, r_n, r_n). \tag{3.47}$$

To compute $S(n, r_n, r_n)$ efficiently, we compute all the subproblem values in table S. Furthermore, let $S'(k, r_k, r_n, b)$, $b \in \{0, 1\}$, be a sum over all length-k relevance sub-paths ending with $z_k = b$, with exactly r_k relevant segments, and normalization factor r_n:

$$S'(k, r_k, r_n, b) = \sum_{z_1,\ldots z_{k-1}, z_k = b, \sum_{i=1}^{k} z_i = r_k} \prod_{i=1}^{k} e^{z_i \left[\xi(x_i, \mathbf{w_r}, \bar{\theta}) + \frac{1}{r_n} \mathbf{w_y} \cdot \sum_{i=1}^{n} z_i f(x_i)\right]}. \tag{3.48}$$

Obviously,

$$S(k, r_k, r_n) = S'(k, r_k, r_n, 0) + S'(k, r_k, r_n, 1). \tag{3.49}$$

$S'(k, r_k, r_n, b)$ can be written recursively in terms of the "smaller" subproblems:

$$S'(k, r_k, r_n, 0) = \begin{cases} 0, & r_k > k \vee r_k > r_n \\ S(k-1, r_k, r_n) = S'(k-1, r_k, r_n, 0) + S'(k-1, r_k, r_n, 1), & o.w. \end{cases} \tag{3.50}$$

$$S'(k, r_k, r_n, 1) = \begin{cases} 0, & r_k > k \vee r_k > r_n \\ S'(k-1, r_k - 1, r_n, 0)\, e^{\mathbf{w_r} g(x_k) + \frac{1}{r_n} \mathbf{w_y} f(x_k)} & \\ + S'(k-1, r_k - 1, r_n, 1)\, e^{\mathbf{w_r} g(x_k) + \theta + \frac{1}{r_n} \mathbf{w_y} f(x_k)}, & o.w. \end{cases} \tag{3.51}$$

Thus, the whole dynamic programming table S' (and therefore S) can be computed starting

from smaller problems, with the following initialization:

$$
\begin{aligned}
S'(1,0,r_n,0) &= 1 \\
S'(1,1,r_n,0) &= 0 \\
S'(1,0,r_n,1) &= 0 \\
S'(1,1,0,1) &= 0 \\
S'(1,1,r_n,1) &= e^{\mathbf{w_r} g(x_1) + \theta_0 + \frac{1}{r_n} \mathbf{w_y} f(x_1)}, \quad r_n > 0
\end{aligned}
$$

The time complexity of this algorithm is therefore $O(Cn^3)$, where C includes the computation of dot product between weight vectors and feature vectors ($\mathbf{w_r} g(x_k)$ and $\mathbf{w_y} f(x_k)$).

Derivatives of $\varphi(y|\mathbf{x})$ with respect to $\nu \in \{\mathbf{w_y}, \mathbf{w_r}, \theta, \theta_0\}$ are computed in a similar fashion.

Chapter 4

Experiments

In this chapter, we present the experiments and results of our method.

First, we describe the three datasets that we use in our experiments. Two synthetic datasets are created from the MIPT terrorist knowledgebase, combining the summaries of terrorist events with the background information on the terrorist organizations, and other events attached to the same terrorist group. They are created in a way to exhibit the relevance structure of a document, and yet to reflect the typical flow of a newspaper article. The classification task is to recognize the method used in a terrorist attacked. The third dataset that we use is a polarity dataset of Pang and Lee [23]. It contains movie reviews labeled as positive or negative (binary classification). While the polarity dataset does not contain any relevance labels, the terrorist datasets are created in a way that the relevance is known, and can be compared against the relevance assigned by our method.

Second, we describe the baselines that we compare against. We use three baselines and one oracle classifier. We use log-linear and SVM classifiers in a relevance-oblivious setting as baselines. They use bag-of-words document representation. Log-linear classifier is the baseline derived from our model, i.e., it represents the labeling part of our model, assuming that every segment is relevant. We also include the majority baseline for the comparison. In the case of terrorist datasets, we compare our model against the oracle classifier, which accesses the "true" relevance information that is present due to the way the documents are created. It is, basically, the bag-of-words classifier applied only to "truly" relevant segments.

A Palestinian man crossed an open desert section of the Egyptian border into Israel, hitched a ride from an Israeli motorist, and then blew himself up inside a bakery in Eilat, killing three people. The al-Aqsa Martyrs Brigade and Palestine Islamic Jihad claimed joint responsibility for the attack. Al-Aqsa Martyrs Brigade is an active terrorist organization committed to the creation of a Palestinian nation-state. The brigade is comprised of an unknown number of small militias, or cells. While never officially recognized by al-Fatah or its former leader Yasir Arafat, al-Aqsa Martyrs Brigade is predominantly comprised of terrorists who also belong to al-Fatah. There have been reports that Arafat approved payments to al-Aqsa Martyrs Brigade. In 2000, the brigades began to operate separately from al-Fatah and have been a significant factor in the current intifada. Al-Aqsa Martyrs Brigade primary tactics are suicide bombings and firearms attacks. While the groups primary objective is to forcibly remove Israelis from the West Bank, Gaza Strip, and Jerusalem, the group also targets civilians and soldiers in Israel.

Table 4.1: An event summary (first paragraph) augmented with background information (second and third paragraphs). Both segments are extracted from the MIPT knowledgebase. The label for this text (suicide) is based on the original summary given in the first paragraph.

Third, we describe features that we use in our experiments. We always use word unigrams as basic features for both relevance and label classifier components. Our method is orthogonal to the choice of the basic feature set, since we are interested in adding the relevance structure to the model, rather than extending the feature set. However, we introduce additional featured for relevance classifier, which are specifically designed to capture the relevance information.

Then, we describe the details of the training procedure and parameter tuning. In order to train the model properly and efficiently, we need to initialize parameters, tune the learning rate, define early-stop criterion, and tune regularization parameters. In addition, we exploit random restarts and model averaging.

Finally, we describe the experiments that we performed. We present and discuss the results of the baselines and the two implementations of our method (CBN and CRF) on the three datasets. We apply the relevance-aware method on both paragraph and sentence level.

4.1 Experimental Set-Up

4.1.1 Datasets

Our first experiment focuses on the task of topic classification. For this experiment we use the MIPT terrorism knowledgebase.[1] This collection contains summaries of terrorist events written by RAND analysts and background articles about organizations, political situation, etc. The event summaries in this collection are short (77.9 words, 4.4 sentences) containing only the key facts about events. In addition to documents, this collection also contains a database that captures certain attributes of each event. In fact, one of the entries in this database records the type of tactic used in the event. The tactic field has eight values, such as kidnapping, bombing and suicide. To extract the value of this field automatically from the event summary, we can employ eight-way classification.[2] Our training, development and testing sets contain 579, 151 and 185 articles, respectively.

Using the MIPT collection, we created two datasets. The first dataset (MIPT1) is used as a "sanity check" for model behavior. Documents in this dataset are constructed by concatenating a pair of texts from the collection that have distinct tactic values. The label assigned to the new text is equal to the label of the first document in the concatenation. To apply our model to this dataset, we treat the two parts of the concatenated document as segments and represent them by unigram features. In this set-up, the location of a segment uniquely determines relevance. Therefore, the hidden variable associated with a segment has a single feature – location. If our relevance model works correctly, it should assign correct values to the hidden variables and match the performance of a classifier that is trained only on the original documents from MIPT.

The second dataset (MIPT2) derived from the MIPT knowledgebase evaluates our model in a more realistic scenario. Starting with an original MIPT summary of a terrorist event, we expand it with background information from the MIPT knowledgebase including the terrorist group profile, the nature of the local conflict and political situation. An exam-

[1] `http://www.tkb.org/`

[2] Classification is more suitable for identifying the type of tactics used than information extraction because the tactics type may not be specifically mentioned in the text. Multiple cues in the document — victims, location, group involved – contribute to determining the tactic type.

ple of such story is shown in Table 4.1. These extended stories mimic in structure typical newspaper reports which supplement event descriptions with background material. The average length of the generated documents is 342.8 words. The ratio between the original summary and added background material is 0.29. In contrast to the first dataset, segments are combined in random order. One advantage of using these automatically constructed stories is that we know which segments are relevant. We can use this relevance information for evaluation purposes comparing automatically predicted relevance against ground truth. The classification performance on the original summary provides an upper bound on the accuracy of the relevance based model.

The third datset (Polarity) is the polarity dataset of Pang and Lee [23]. It contains 1000 positive and 1000 negative reviews from 312 authors, with the maximum of 20 reviews per author.

4.1.2 Baselines

The relevance oblivious baseline derived from our model is a bag-of-words log-linear classifier. This classifier is the label prediction component of our model operating under the assumption that all segments are relevant: $z_i = 1$ for all $i = 1, \ldots, n\}$ (compare to Eq. 3.3):

$$P(y|\mathbf{x}; \mathbf{w_y}) = \frac{e^{\mathbf{w_y} \cdot \sum_{i=1}^{n} f(x_i)}}{\sum_{y'=y_1}^{y_m} e^{\mathbf{w_{y'}} \cdot \sum_{i=1}^{n} f(x_i)}} \tag{4.1}$$

Alternatively, we can write it as

$$P(y|\mathbf{x}; \mathbf{w_y}) = \frac{e^{\mathbf{w_y} \cdot f(\mathbf{x})}}{\sum_{y'=y_1}^{y_m} e^{\mathbf{w_{y'}} \cdot f(\mathbf{x})}} \tag{4.2}$$

where $f(\mathbf{x})$ is a document feature vector ($f(\mathbf{x}) = \sum_{i=1}^{n} f(x_i)$).

Since there are no hidden variables in the baseline, there is no summation, and, therefore, training and inference do not require sampling.

We also include in the comparison a state-of-the-art classification algorithm based on Support Vector Machines, that again does not use relevance data. We employ Joachims' [10] SVM^{light} package for training and testing with all parameters set to their default values.

In addition, we include the majority baseline and the relevance oracle that is trained and tested only on relevant segments.

4.1.3 Features

Relevance Prediction Relevance features (g(x)) encode our intuition about the properties of the segment that contribute to its relevance. Clearly, these features would be application specific. For instance, when processing newspaper articles we may want to focus on segments that describe the current event. Features such as position in a document, tense of the verbs and the presence of dates may help to distinguish the description of the current event from the background information. We may also want to take into account the relevance of preceeding segments since these decisions are clearly correlated. On MIPT1 dataset we use segment position as a relevance feature. On MIPT2 dataset, we use word unigrams, date mentions, verb tenses, and the location of the topic change obtained by applying min-cut segmentation model [16]. On the polarity dataset, we use word unigrams as relevance features.

Label Prediction The features we use for label prediction (encoded as f(x)) are based on the bag-of-words approach. Namely, we use word unigrams in all the experiments. In practice, there is no limitation in what features will be used. However, we do not employ rich set of features, since we are exploring the orthogonal approach in this work.

4.2 Training and Tuning

In this section, we explain the practical details of the training process. We describe the important issues regarding gradient descent method and regularization parameters tuning.

4.2.1 Initialization

We implemented several initialization schemes for the parameters. In the standard scenario parameters are drawn uniformly from $[0, MAX]$. Alternatively, we allow both positive and

initialization scenario	accuracy
ALL ZEROS ($MAX = 0$)	81.1
$MAX = 10^{-10}$	81.6
$MAX = 10^{-9}$	81.1
$MAX = 10^{-8}$	81.1
$MAX = 10^{-7}$	81.1
$MAX = 10^{-6}$	81.6
$MAX = 10^{-5}$	81.6
$MAX = 10^{-4}$	81.6
$MAX = 10^{-3}$	81.6
$MAX = 10^{-2}$	81.1
$MAX = 10^{-1}$	81.6
$MAX = 1$	81.1
POSITIVE+NEGATIVE ($MAX = 10^{-10}$)	81.1
INIT FROM BASELINE	81.6

Table 4.2: Accuracy on the development set using different initialization scenarios.

negative values, i.e. $[-MAX, MAX]$. Finally, we also try to initialize the parameters of the Relevance-Aware model with the baseline model.

We show that the initialization of parameters does not play significant role in the training process. In our experiment, we try these initialization schemes under the same training conditions. [3] We use the MIPT2 dataset and CBN implementation in this experiment. The development accuracy for each scenario is shown in Table 4.2. Obviously, the magnitude of the initial values is not important. A possible explanation is that the parameters are "dragged" significantly in the direction of the gradient in the first iteration of the gradient descent algorithm. Unimportant parameters with large value are pulled towards zero by the regularization term. In other words, the issue with local minima might come only later in the training process. The fact that only labeling part of the model is affected in the beginning of training supports this assertion. Namely, all segment relevances are initially around 0.5.

In all subsequent experiments, we use initialization with positive values ($MAX = 10^{-10}$).

[3] Regularization constants and initial learning rate are fixed, as well as the order of training examples for online updates.

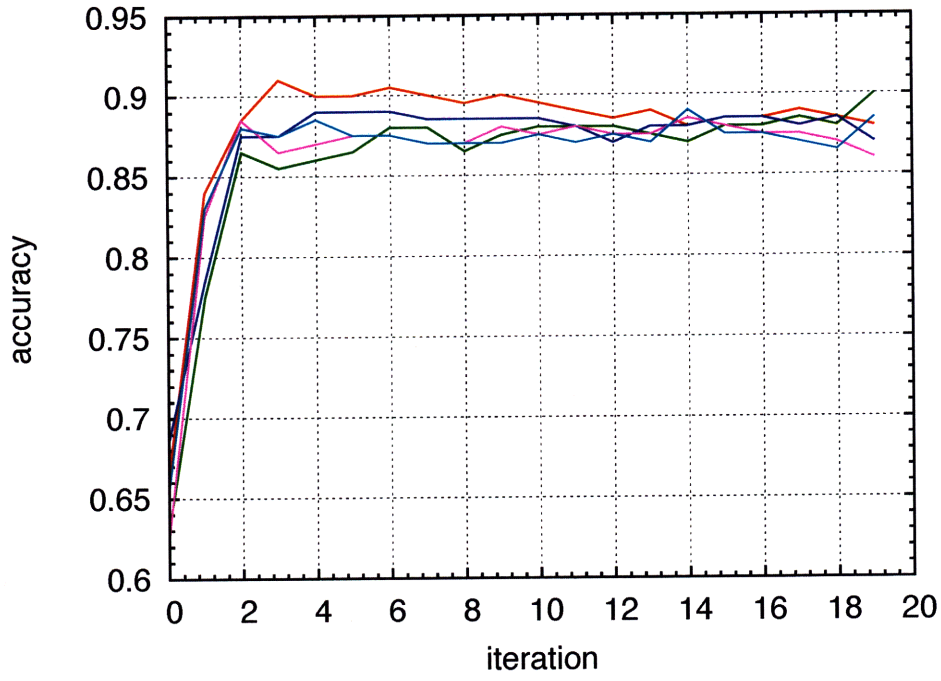

Figure 4-1: Accuracy as a function of the training iteration, displayed for five random trials.

4.2.2 Random Restarts

As shown in 4.2.1, randomization in the initialization does not affect the result significantly. However, we randomly change the order of training examples in each iteration. Different random restarts may lead to different optimum points. The change of the development set accuracy during training process is shown in Figure 4-1 for five random trials. [4] In order to make the results stable, we perform five trials of training and take average accuracy in all the experiments.

4.2.3 Learning Rate Tuning

In gradient descent, parameters are typically updated by the following rule:

$$\nu = \nu - \eta \frac{\partial L}{\partial \nu}, \tag{4.3}$$

[4] This plot is obtained on the polarity dataset, using CBN method.

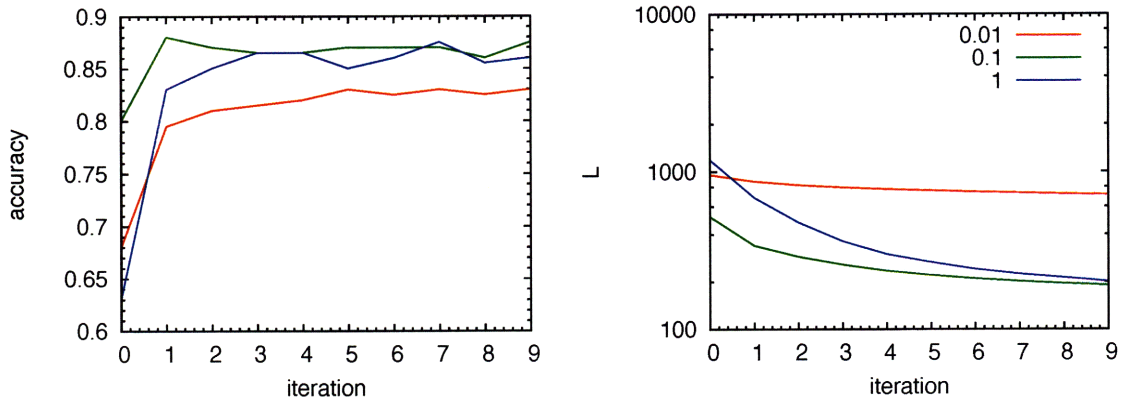

Figure 4-2: Accuracy (left) and objective (right) as a function of training iteration, displayed for three initial learning rates that are apart from each other by a factor of 10 (0.01, 0.1, 1).

where L is the objective function to minimize, and η is a learning rate [2, 5]. After setting initial learning rate η_0, we decrease it after each iteration i according to the following rule [7]:

$$\nu_i = \frac{\nu_0}{i+1}. \tag{4.4}$$

We tune initial learning rate to the value that minimizes the objective function L after t iterations. In our experiments, we show that it is sufficient to set a learning rate that is up to a factor of 3 greater or smaller than the best one. This is illustrated in Figures 4-2 and 4-3. [5] Therefore, we try only couple of predefined values (0.1, 0.3, and 1) and extend search to larger or smaller values if necessary (e.g., if η=1 is the best of the three predefined choices, we also try values higher than 1, until the objective function starts to increase again). We use $t = 2$. In most cases, there is no difference among choices $t \geq 2$, and the rate that initially decreases L by greatest value will keep that margin in later iterations.

4.2.4 Early-Stop Procedure

We employ an early-stop procedure to choose when to stop with gradient descent iterations. We analyze three possible criteria:

[5] This plots are obtained on polarity dataset.

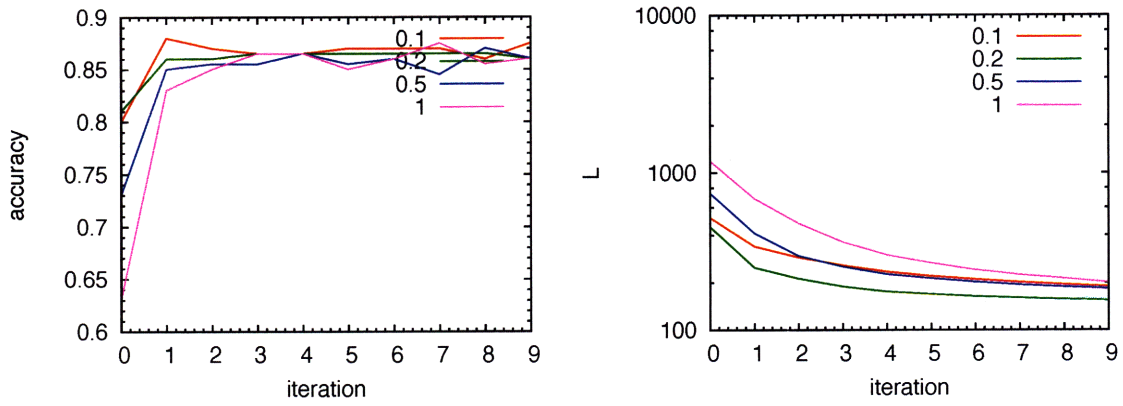

Figure 4-3: Accuracy (left) and objective (right) as a function of training iteration, displayed for four initial learning rates that are apart from each other by a factor less than 3 (0.1, 0.2 0.5, 1).

- After k iterations
- When the relative decrease in the objective function $((L_{prev} - L)/L_{prev})$ is smaller than a threshold τ
- When the relative decrease in the training log-likelihood is smaller than a threshold τ

We tune k and τ using a heldout dataset. Although, for each training scenario (dataset + model + regularization constants), the results using these three criteria are almost identical, we found that iterating until the relative decrease in the training log-likelihood drops below $\tau = 0.04$ is a good universal rule across all scenarios.

4.2.5 Averaging

In each training iteration of the stochastic gradient descent, parameters are updated on each training example. In one update, parameters are changed according to the gradient computed using only one example. To reduce the influence of the most recent updates, we compute the average model of the last k iterations, i.e. it is an average of models obtained after each update in these k iterations. We use $k = 1$ in all the experiments.

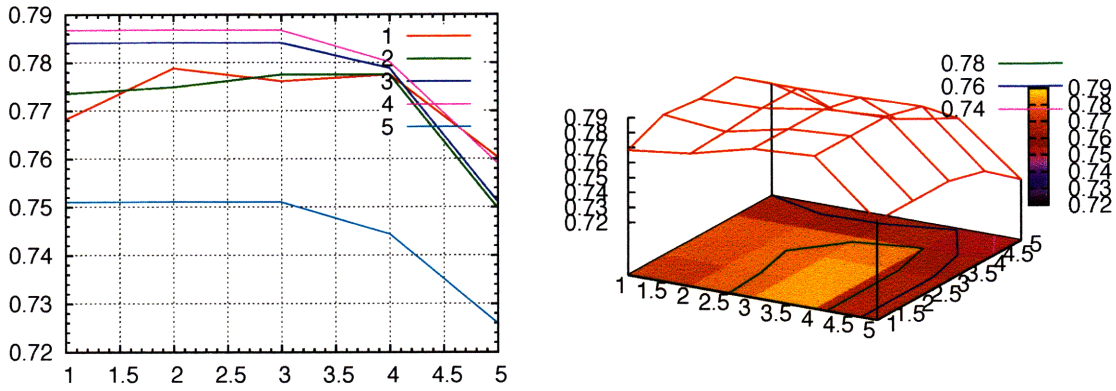

Figure 4-4: Accuracy as a function of regularization constants. In the left figure, each line corresponds to a fixed value of C_y parameter, while x-axis corresponds to C_r parameter. In the right figure, x-axis (the closer one) and y-axis correspond to C_y and C_r, respectively. Note that the numbers in plots that correspond to regularization constants do not represent values. Instead, they map to some values, such that a higher number corresponds to a higher regularization constant. This plot is obtained on the MIPT2 dataset using CBN model.

4.2.6 Regularization

Our model differs from the standard log-linear model in that it has two regularization parameters (C_r and C_y). Although we could have the same regularization parameter over relevance and labeling parameters, there is no reason to put such a constraint. In fact, the best relevance-aware models in our experiments are obtained through training with significantly different constants. However, it comes at a cost of a more complex tuning procedure.

We develop a supervised procedure for tuning regularization parameters. We combine several predefined values for C_y and C_r, train a model using each pair of values, and test it on the development set. We generate plots as in Figure 4-4 as a visual aid. From the results, we choose a "2D subregion" that spans the best models (possibly choosing to explore an area out-of-the-scope of the predefined values). Then, we use a "finer" division of the subregion to localize the best parameters. Typically, we repeat this procedure two or three times until the best parameters are found. If there is a span of good values for the parameter, we choose the largest one in order to have the simplest model.

	MIPT1
MAJORITY	67.6
RELEVANCE-OBLIVIOUS (LOG-LINEAR)	68.6
RELEVANCE-OBLIVIOUS (SVM)	67.6
RELEVANCE-AWARE CBN	84.3
RELEVANCE-ORACLE	83.8

Table 4.3: Accuracy on the MIPT1 dataset

4.3 Results

In this section, we give results of our experiments. In the tables, RELEVANCE-AWARE denotes our method (CBN or CRF implementation). RELEVANCE-OBLIVIOUS denotes a relevance-oblivious baseline (LOG-LINEAR or SVM). MAJORITY denotes the majority baseline, and RELEVANCE-ORACLE denotes the method that has access to the "correct" relevances, and use only relevant segments. Unless stated otherwise, **averaging** scheme for combining features is used, in order to utilize efficient CRF implementation.

4.3.1 Results on MIPT1 dataset

Table 4.3 summarizes the performance of our method, the baselines and the relevance oracle on the MIPT1 dataset.[6] As we expect, two relevance-oblivious baselines achieve results close to majority. The presence of the second paragraph, which in most cases belongs to the majority class, outweights the influence of the first (relevant) paragraph. In a more balanced training, we would expect a result close to random. At the same time, the performance of our model is close to that of the relevance oracle (statistically insignificant difference). This is not surprising given the perfect detection of relevant paragraphs by our model on this dataset.

[6]There is one dominant class that is assigned to 67.6% of the documents, thus dictating the high majority baseline score.

	MIPT2-PAR	MIPT2-SEN
MAJORITY	67.6	67.6
RELEVANCE-OBLIVIOUS (LOG-LINEAR)	77.3	77.3
RELEVANCE-OBLIVIOUS (SVM)	78.4	78.4
RELEVANCE-AWARE CBN	83.2	79.5
RELEVANCE-AWARE CRF	82.7	75.5
RELEVANCE-ORACLE	83.8	83.8

Table 4.4: Accuracy on the MIPT2 dataset. PAR stands for the paragraph level, while SEN stands for the sentence level.

4.3.2 Results on MIPT2 dataset

On the second MIPT dataset, our model continues to deliver significant performance gains over the baselines (Table 4.4). It also matches the performance of the relevance oracle. What is interesting, however, is that these results are achieved despite a low relevance detection rate. The model identifies 333 segments as relevant, in comparison to 185 original summary segments available in the test corpus. Our manual analysis of this data reveals that the relevance prediction component eliminates segments in a conservative way. Only segments that contain "confusing" words are eliminated. If a segment does not interfere with a target classification, it tends to be marked as relevant. These results suggest that that human annotation of relevance may not necessarily overlap with an automatically induced relevance assessment.

We explore the optimal segment granularity for relevance-aware models. The goal is to find the right balance between the representation power of the model and the complexity of the model with respect to its parametrization. We hypothesize that sentence level analysis may not predict sufficient information for relevance assessment. In fact, our features are more meaningful for larger units of text. As an empirical confirmation, the accuracy of our model on MIPT2 data drops to 79.5% (CBN implementation) when training and testing are performed using sentence level segments. On the other hand, the result of the CRF implementation is worse than the baseline. The inspection shows that the relevance weights tend to be negative. The CRF variant is, in this case, more difficult to train, which may stem from the fact that the implementation model is normalized globally. In the case of

	Polarity-PAR
RELEVANCE-OBLIVIOUS (LOG-LINEAR)	85.5
RELEVANCE-OBLIVIOUS (SVM)	86.3
RELEVANCE-AWARE CBN	85.1
RELEVANCE-AWARE CRF	84.8

Table 4.5: Accuracy on the polarity dataset on the paragraph level.

paragraph level, this might not be a problem, because of the small and fixed number of paragraphs in this dataset.

In another experiment, we provide our relevance features to the baseline model. However, the result does not change. Also, by using only unigram features as relevance features, the result of the relevance-aware method decreases insignificantly. This eliminates the possibility that the relevance-aware method gains only from the relevance features, and shows that the relevance structure helps.

We also compare the result of the CBN implementation using exact training and inference with the sampling method. The small number of paragraphs in MIPT2 dataset allows us to compute all the derivatives exhaustively. The accuracy of the exact method on paragraph level is 83.8%, and is not significantly different from the results obtained by the sampling method.

4.3.3 Results on Polarity dataset

On the polarity dataset, our method does not improve over the baseline. The results on the paragraph level are given in Table 4.5. By looking at the training process, we can see that the relevance-aware method gives best result when regularization constant on relevance parameters is extremely high. This is shown in Figure 4-5. Essentially, the model works better on this dataset without relevance information that with it. The high value C_r drives relevance parameters towards zero, making each segment roughly 50% relevant.

If we reduce C_r, our method assigns some non-trivial probability of relevance to each paragraph. However, the accuracy drops by 2%. Table 4.6 shows one correctly labeled negative review with the attached relevance. In this case, the last paragraph, which clearly

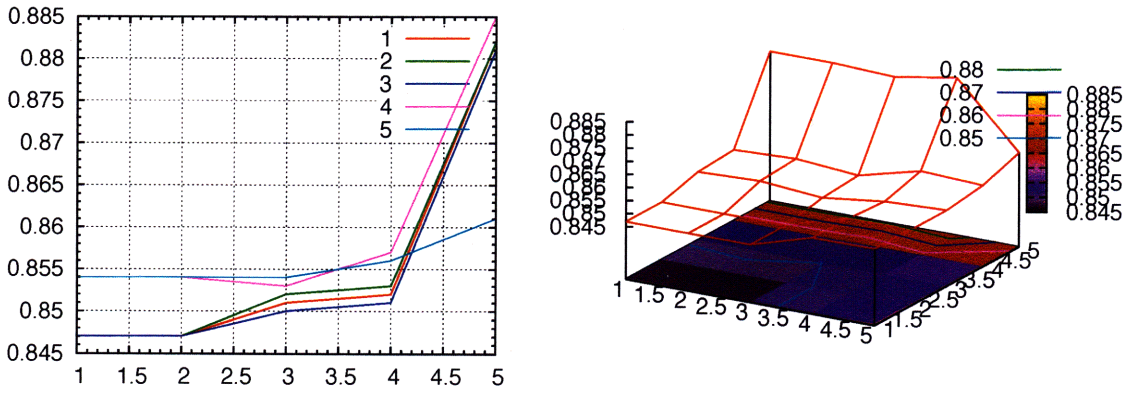

Figure 4-5: Accuracy as a function of regularization constants. In the left figure, each line corresponds to a fixed value of C_y parameter, while x-axis corresponds to C_r parameter. In the right figure, x-axis (the closer one) and y-axis correspond to C_y and C_r, respectively. This plot is obtained on the polarity dataset using CRF model in order to show the inability of training a relevance model that helps target classification. Note that the numbers in plots that correspond to regularization constants do not represent values. Instead, they map to some values, such that a higher number corresponds to a higher regularization constant.

exhibits the sentiment of the review, gets the highest relevance score (0.85). On the other hand, the third paragraph, that only describes the plot of the movie, gets the lowest relevance score. The other two paragraphs mostly describe the plot, but also contain some sentiment information. This suggests the potential application of our method to sentiment-driven summarization. However, in the current scenario, relevance assignments in many reviews are counterintuitive, and do not help the classification. We conclude that the relevance part of our model overfits, i.e., it is not capable of tuning the relevance weights for the better classification performance on the test set. This is not surprising given the small amount of training data and many features in the unsupervised learning scenario. In other words, there are no enough significant relevance features that constitute strong repeating patterns correlated with the final classification task.

Our results are, in fact, similar to those of Pang and Lee [23] on the same dataset, since they do not show improvement over the SVM baseline, even though they train relevance model in a supervised fashion. First of all, note that it is not guaranteed that what humans regard relevant contains information that correlates well with the final classification task.

For example, in our experiments, we do not recover any relevance model that is beneficial. Second, it is specific for reviews to be coherent, which may explain the lack of improvement over the baseline. Namely, an overall positive review will probably be positive throughout the most of its content, i.e., "happy" users will likely neglect some negative aspects of a product. Even if there is a negative irrelevant part of the review, it may not be enough to outweight the positive features. The analogue holds for negative reviews. Therefore, looking into the document structure may not surpass the prediction power of the document as a whole. Finally, the joint supervised model of McDonald et al. [19] also do not achieve a consistent improvement across different dataset, which reinforces our assertion.

#	*paragraph*	P_{rel}
1	Whether you like the Beatles or not, nobody wants to see the Bee Gee's take on some of the Fab Four's best known songs. Well, maybe that's not true . . . maybe you're curious, the way you have to look in your hanky after you blow your nose. You just have to know how bad bad can be. If that's the case, rejoice, because it was twenty years ago today (or so) that SGT. PEPPER'S LONELY HEARTS CLUB BAND was released (unleashed?) to the world, and thanks to our modern technological advances, you can find this retched piece of filmmaking on VHS.	0.35
2	Derived from the lyrics of various Beatles' songs, SGT. PEPPER'S tells the story of the fictitious band made popular from the song (and album) of the same name, released in 1967. Of course, the movie was made eleven years later, the Gibbs three have become the Lonely Hearts, Peter Frampton is the one and only Billy Shears, and aside from being about a rock band, the story doesn't correlate to the song at all. And oh, what joy - we're the lovely audience they'd like to take home with them. I don't think so. But at least these characters are actually *people* in a Beatles' song, whereas other characters such as Strawberry Fields (Sandy Farina) gets her name from a song about a *place* called Strawberry Fields. The debate over this is really quite futile when it comes down to it, because all the film really has to offer is a feast of horrid cover tunes, embarrassing cameo appearances (George Burns?! Steve Martin?!? ALICE COOPER?!?!), and UGLY 70's fashion and faces.	0.41
3	The plot is a bit unclear. People with bad 70's hair run around in leisure suits engaging in music video sequences that look like a sick and twisted world of make-believe from an overly demented Mr. Rogers. Mean Mr. Mustard (Frankie Howerd) somehow gets hold of all Billy & Co.'s instruments, calling Dr. Maxwell Edison (Steve Martin with a silver hammer and an out-of-key singing voice), the Sun King, Marvin Sunk (Alice Cooper!), and a couple of creepy robots to his aid. Supposedly this is a horrible thing? I guess in Heartland, the talent is pretty non-existent. Meanwhile, the Lonely Hearts are off doing the classic "sex, drugs, and rock n' roll" thing, leaving poor Strawberry Fields without her true love, Billy ...	0.24
4	It's movies like this that make ya sit back and ask the unanswerable question, "What the hell were they thinking???" Nobody will ever know, but as a novelty, SGT. PEPPER'S is one to examine. Carol Channing, Robert Palmer, Keith Carradine - they're all here. But why???? Who knows. It's irrelevant. There's as much meaning to be found here as there is to be found in your belly-button lint, although the latter may be more interesting. With the recent onslaught of 70's nostalgia in the movie world (THE ICE STORM, BOOGIE NIGHTS, reissues of the STAR WARS trilogy, etc.), let's pray this doesn't get a special 20th anniversary, second-chance in theaters. In the words of Paul McCartney, live and let die. In fact, bury this one while you still can.	0.85

Table 4.6: A movie review with assigned relevance probabilities of paragraphs.

Chapter 5

Conclusions and Future Work

In this work, we presented a method for text classification that takes into account segment relevance. In contrast to the previous work, which assumes that relevance annotation is given, we represent relevance as a hidden variable of a target classifier. Relevance and label predictions are performed jointly, optimizing the relevance component for the best result of the target classifier.

Our method yields significant empirical improvements over relevance-oblivious classifiers on the MIPT1 and MIPT2 dataset. However, it does not show improvement on the Polarity dataset. There are multiple facts that contribute to explaining this result. First, the model may overfit. McDonald et al. [19] suggest that their joint model probably overfits when determining document level label, because training error decreases much faster than in the case of a Document-Classifier. They get only small improvement on document accuracy. Second, unsupervised training is difficult in general. There are many parameters supporting the hidden layer (which can cause of overfitting). Also, there are multiple local optima, and it is easy to get stuck in one of them. Third, mistakes in relevance prediction influence label prediction. Sensitive to relevance information stems from the fact that classification relies only on the relevance portion of the text. Although marginalization "gives chance" to all segments, the weights put on highly irrelevant segments will be very small, and they will not influence the result. Therefore, we might have "almost" hard assignment of relevance based on relevance classification component, and, at test time, irrelevant segments are not given chance. This is solved to some extent in the work of Patwardhan and

Riloff [24], where information extraction is also performed on irrelevant segments, but only using the most common patterns that have high extraction confidence. That way, they compensate mistakes of the relevance classifier. McDonald et al. [19] model the dependencies between document and sentence level labels through features that combine possible values of these labels. That essentially allows the model to learn what is a relationship between labels and distinguish the influence of relevant segments from the influence of irrelevant segments. Finally, binary classification task on Polarity dataset is easier than mutliclass classification task on MIPT datasets, and there are already a lot of evidences in documents that lead to the high accuracy of the baseline classifier. Reviews tend to be coherent, i.e., if a review is positive, it will probably be positive in all aspects, and throughout the whole review (the analogue holds for negative reviews). The lack of predictive relevance features might also be a cause that our method does not outperform the baseline. Introducing such features is a possible extension to our work.

There are several possible improvements of our method that we consider working on in the future. One drawback of our model is that we do not control the number of relevant segments. It can be extended to include regularization on that number, i.e. to encourage a specific number of relevant segments. By introducing probability distribution over the number of relevant segments we can avoid the situation in which the segments are always relevant or always irrelevant. We might also be able to learn a better relevance model. In addition, there might be further gains if the model is trained as a max-margin model [9, 32], i.e. if objective function enforces large margin between the correct label and the second best label.

The problem might be difficult. Pang and Lee only improve over the weaker naïve Bayes baseline, but not over SVM. McDonald et al. get only small and inconsistent improvement of the document level accuracy, while the sentence level accuracy gain is much higher. This implies that the influence is much higher going from document to sentence labels than vice versa, suggesting that there is more benefit in making local decisions knowing its broader context than vice versa. This is intuitive. In document labeling, if a confusing segment is included, there is a good chance that there is enough information in other segments to outweight its influence and still classify the document correctly. In

information extraction, inclusion of a confusing segment has bigger consequences for the performance of the system, since the extraction within that segment is performed based on local featues, without knowing its broader context. For example, the work of Patwardhan and Riloff heavily exploits that direction of influence. In the future, we plan on developing IE variant of our method.

Our current model considers a simple representation of document structure modeling text as a linear sequence of segments. In the literature, however, discourse is frequently modeled using a hierarchical representation such as a Rhetorical Tree Structure [17, 18]. An important avenue for future research is to incorporate more elaborate discourse models into text classification algorithms. This line of research may not only lead to improved classification accuracy, but also shed new light on the representational power of different discourse structures.

Bibliography

[1] Adam L. Berger, Stephen A. Della Pietra Y, and Vincent J. Della Pietra Y. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22:39–71, 1996.

[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.

[3] Michael Collins. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 1–8, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

[4] Koby Crammer and Yoram Singer. Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.*, 3:951–991, 2003.

[5] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

[6] Jacob Eisenstein and Randall Davis. Conditional modality fusion for coreference resolution. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 352–359, Prague, Czech Republic, 2007.

[7] Charles Elkan. Log-linear models and conditional random elds, 2008. Notes for a tutorial at CIKM08.

[8] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, May 1998.

[9] Y. Guo, D. Wilkinson, and D. Schuurmans. Maximum margin bayesian networks. In *Proceeding of UAI*, 2005.

[10] Thorsten Joachims. *Learning to Classify Text Using Support Vector Machines*. Kluwer, 2002.

[11] Terry Koo and Michael Collins. Hidden-variable models for discriminative reranking. In *Proceedings of HLT/EMNLP*, 2005.

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

[13] Pat Langley, Wayne Iba, and Kevin Thompson. An analysis of bayesian classifiers. In *In Proceedings of the tenth national conference on artificial intelligence*, pages 223–228. AAAI Press, 1992.

[14] David D. Lewis. Text representation for intelligent text retrieval: a classification-oriented view. pages 179–197, 1992.

[15] David D. Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. pages 4–15. Springer Verlag, 1998.

[16] Igor Malioutov and Regina Barzilay. Minimum cut model for spoken lecture segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, Sydney, Australia, July 2006. Association for Computational Linguistics.

[17] William C. Mann and Sandra A. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *TEXT*, 8(3):243–281, 1988.

[18] Daniel Marcu. The rhetorical parsing of natural language texts. In *Proceedings of the ACL/EACL*, pages 96–103, 1997.

[19] Ryan McDonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 432–439, June 2007.

[20] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244, 1990.

[21] Alessandro Moschitti and Roberto Basili. Complex linguistic features for text classification: A comprehensive study. In *Proceedings of ECIR*, pages 181–196, 2004.

[22] Keith Noto and Mark Craven. Learning hidden markov models for regression using path aggregation. In *Proceedings of the 24th Annual Conference on Uncertainty in Artificial Intelligence (UAI-08)*. AUAI Press.

[23] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 271–278, Barcelona, Spain, 2004.

[24] Siddharth Patwardhan and Ellen Riloff. Effective information extraction with semantic affinity patterns and relevant regions. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 717–727, 2007.

[25] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[26] Ariadna Quattoni, Michael Collins, and Trevor Darrell. Conditional random fields for object recognition. In *Proceedings of NIPS*, 2004.

[27] Adwait Ratnaparkhi. A simple introduction to maximum entropy models for natural language processing. Technical report, 1997.

[28] Ellen Riloff. Automatically generating extraction patterns from untagged text. In *In Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1044–1049, 1996.

[29] Tobias Scheffer and Stefan Wrobel. Text classification beyond the bag-of-words representation. In *Proceedings of the ICML-Workshop on Text Learning*, 2002.

[30] Sam Scott and Stan Matwin. Feature engineering for text classification. In *Proceedings of ICML-99, 16th International Conference on Machine Learning*, pages 379–388, 1999.

[31] Beth M. Sundheim. Overview of the fourth message understanding evaluation and conference. In *MUC4 '92: Proceedings of the 4th conference on Message understanding*, pages 3–21, Morristown, NJ, USA, 1992. Association for Computational Linguistics.

[32] Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin markov networks. MIT Press, 2003.