Reference Resolution and Its Application to Legal Question Answering

by

Tran Thi Oanh

submitted to
Japan Advanced Institute of Science and Technology
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Supervisor: Professor Akira Shimazu

School of Information Science
Japan Advanced Institute of Science and Technology

March, 2014

Abstract

Natural languages are highly related by references within them. These references bring precious information: the sentences of a discourse could not be interpreted without knowing who or what entity is being talked about. Resolving resolution, therefore, is a very important task in natural language processing research. Of all reference phenomena, the coreference is the most popular phenomenon, and is attracting much research in reference resolution. In this dissertation, we will concentrate on this challenging task - *coreference resolution in general texts*. Moreover, we will also focus on *resolving references in a specific type of texts, i.e. legal texts*. The information on reference resolution not only helps people in understanding texts, but also supports other tasks such as question answering, text summarization, and machine translation. To illustrate one of these benefits, in this thesis, we will also investigate *an application of reference resolution to the task of question answering restricted to the legal domain*.

Most previous research proposed a pairwise approach to solve the task of coreference resolution. The drawback of this approach is that it can allow only one or two antecedent candidates to be considered simultaneously. So, it only determines how good a candidate is relative to the mention, but not how good a candidate is relative to all candidates. Our goal is to investigate another approach which can address this drawback. While coreference resolution in general texts attracts much attention among researchers, the task in legal texts has received very little attention so far. The main reasons are mostly the complex and long legal structures and sentences, specific terms, and especially the lack of language resources (i.e. annotated corpora) in this specific domain. Focusing on this interesting legal domain, this dissertation also aims at building a system which can automatically extract referents for references in real time. This is a new interesting task in the Legal Engineering research. Moreover, the goal of this dissertation also includes building an application of these reference resolvers to a useful question answering system restricted to the legal domain. Particularly, the following three problems are targeted in this research:

- To realize coreference resolution in general texts, we present an empirical study on a listwise, which can address the drawback of the previous approach. This approach exploits a listwise learning-to-rank method which considers all antecedent candidates simultaneously, not only in the resolution phase but also in the training phase. Experimental results on the corpora of SemEval-2010 shared task 1 show that the proposed system yields a good performance in multiple languages when compared to previous participating systems as well as a baseline pairwise system using the ranking support vector machine as the learning algorithm. In comparison to the best participating system SUCRE, which uses the Decision Tree algorithm with best-first clustering strategy, the proposed system achieves comparative performance.

- For the task of reference resolution in legal texts, different from previous work that only considered the referent at the document targets, this work focuses on resolving references to the sub-document targets. Referents extracted are the smallest

fragments of texts in documents, rather than the entire documents that contain the referenced texts. Based on the structures of references in legal texts, we propose a four-step framework to accomplish the task: mention detection, contextual information extraction, antecedent candidate generation, and antecedent determination. We also show how machine learning methods can be exploited in each step. The final system achieves 80.06% in the F1 score for detecting references, 85.61% accuracy for resolving them, and 67.02% in the F1 score on the end-to-end setting task on the Japanese National Pension Law corpus.

- This dissertation also presents a study aimed at exploiting reference information to build a question answering system restricted to the legal domain. Most previous research focuses on answering legal questions whose answers can be found in one document[1] without using reference information. However, there exist many legal questions, which require answers extracted from connections of more than one document. The connections between documents are represented by explicit or implicit references. To the best of our knowledge, this type of questions is not adequately considered in previous works. To cope with them, we propose a novel approach which allows to exploit the reference information between legal documents to find answers to these legal questions. This approach also uses the requisite-effectuation structures of legal sentences and some effective similarity measures based on legal terms to support finding correct answers without training data.

The contribution of this dissertation includes linguistic and computational aspects. Considering the linguistic viewpoint, our research helps in interpreting the sentences of any discourse. In the computational viewpoint, our research proposes effective solutions for linguistic problems using machine learning approaches.

Keywords: reference resolution, coreference resolution, legal texts, question answering, pairwise approach, listwise approach, learning-to-rank, logical structure, requisite-effectuation structures, mention detection, JNPL corpus.

[1]The term '*documents*' corresponds to articles, paragraphs, items, or sub-items according to the naming rules used in the legal domain.

Acknowledgments

First of all, I would like to express my special thanks to my great supervisor, Professor Akira Shimazu of the Natural Language Processing laboratory, at the School of Information Science, of JAIST, for the patient guidance, encouragement and advice which he has offered me throughout my study time. He always gave me more enthusiasm and pushed me to do better in my research topic. He transmitted to me much invaluable knowledge in not only the way to formulate a research idea, to write a good paper, etc. but also the vision and much useful experience in academic life. I feel really lucky and so proud to be one of his students.

I would also like to express my special thanks to Associate Professor Kiyoaki Shirai for his useful and valuable discussions and comments during my study period.

I also would like to express my gratitude to Associate Professor Nguyen Le Minh for many helpful discussions with him on conducting the research. He has given me many valuable comments, experience and constant support in my study since my early days at JAIST.

My sincere thanks also go to Professor Ho Tu Bao for his support and encouragement during my life at JAIST and my research, especially for my sub-theme study.

I would like to thank committee members, consisting of Professor Takenobu Tokunaga at the Tokyo Institute of Technology, Professor Satoshi Tojo, Professor Ho Tu Bao, and Associate Professor Kiyoaki Shirai at JAIST, who have been supportive beyond the call of duty. They have reviewed my dissertation and provided valuable insight. My dissertation is improved very much through valuable comments.

I would like to express my appreciation to my former supervisor, Associate Professor Ha Quang Thuy, and my former co-supervisor, Associate Professor Le Anh Cuong, for their guidance to my master's thesis and my bachelor thesis. Professor Ha Quang Thuy is also a leader of a scientific research group at the University of Engineering and Technology, the Vietnam National University in Hanoi, where I accquired much knowledge through weekly seminars.

I would like to thank the English Language Education for Science, Technology and Engineering (CELESTE) for their help in proofreading and correcting errors in my papers. I have learned a lot from them.

I sincerely thank all my friends and colleagues who always supported me in times of need. I greatly appreciate all members of the Shimazu and Shirai laboratory for their help and contributions in building a wonderful and supportive academic environment. I also would like to thank many Vietnamese friends at JAIST for the good times we spent together over four years.

I also deeply acknowledge the Monbukagakusho for financial support during my PhD course at JAIST through a scholarship funded in the form of the Japanese Ministry of Education, Culture, Sports, Science, and Technology. I also would like to thank the Grant-in-Aid for Scientific Research, Education and the Research Center for Trustworthy e-Society, JAIST Research Grants, and the JAIST Overseas Training Program for

3D Program Students in supporting me to do the research and attending international conferences for presenting my work. I also would like to thank all JAIST staff for its kind support in many official procedures.

Last, but not least, I would like to express my gratitude to my sweet family, which is really my biggest motivation. They always give me encouragements, care, love, and support in my daily life. They are endless sources of inspiration for me to move forwards, so this thesis is dedicated to them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Reference resolution is a task which consists of determining which entities are referred to by which linguistic expressions. Of all reference phenomena, the coreference phenomenon is the most popular one and is attracting much research on reference resolution. When a referent is first mentioned in a discourse, we say that a representation for it is evoked into the model. Upon a subsequent mention, this representation is accessed from the model. Figure 1.1 illustrates the operations and the relationships between them.

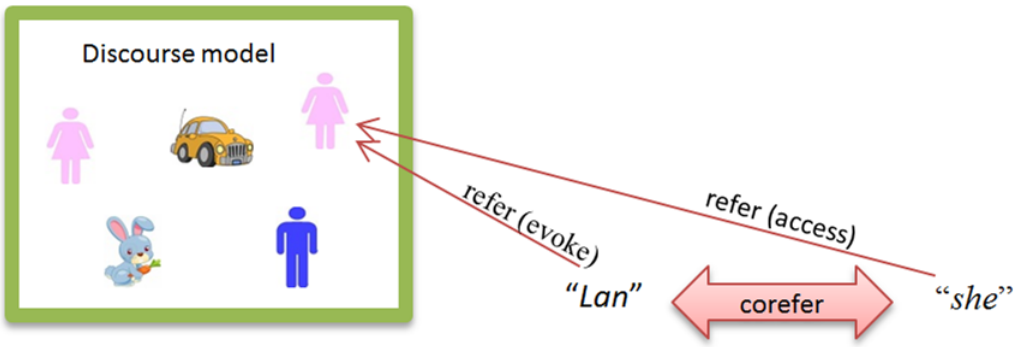

Figure 1.1: Reference operations and relationships with respect to the discourse model.

Coreference resolution has been a core research topic in NLP. In order to derive the correct interpretation of texts, or even to estimate the relative importance of various mentioned subjects, pronouns and other referring expressions need to be connected to the right individuals. In an example sentence '*Lan told me that she would come to the party*', '*she*' and '*Lan*' are most likely referring to the same person, in which case they are coreferent. This type of reference is very typical in the sense that we usually first introduce a person, a location, or a discussion topic by using a relatively long or detailed description, such as a definite description. However, later mentions are briefer, and frequently ambiguous. In the above example, the mention '*she*' may refer to another person rather than '*Lan*' depending on a given context. The resolution of these references, therefore, is very important to the correct understanding of texts. Besides, it also has important applications in areas such as question answering [38, 70], machine translation [87], automatic summarization [4, 121] and named entity extraction [40].

Coreference resolution has a long research history. Algorithms for the problem of pronoun resolution have been developed since the seventies, such as the Hobbs algorithm [42] and the Centering algorithm [14]. They were primarily based on linguistics information. While the early methods incorporated a lot of domain and linguistic knowledge, the newer methods have shown an inclination towards applications of machine learning-based approaches since the mid-to-late 90s. For example, a method uses simple statistic naive bayes-based model [34], methods using decision trees [104] and conditional random fields [66]. Generally speaking, learning-based coreference resolution approaches can be classified into three important classes, namely, the mention-pair model [77, 91], the entity-mention model [62, 126], and ranking models [25, 74]. In the first two classes, each antecedent candidate is resolved independently from the other candidates. So the models could not determine the best candidate in the relation with the other candidates. To address this drawback, ranking models were proved to be useful solutions [25, 74]. However, this weakness is not fully solved to the extent that these models cannot examine all antecedent candidates at the same time. By default, it is strongly assumed that candidates or pairs of candidates are generated independently and identically distributed, and the trained models will be biased towards mentions with more candidates. Another problem is that the objective of learning is formalized as minimizing classification errors of candidates or pairs of candidates, rather than minimizing ranking errors of candidates globally.

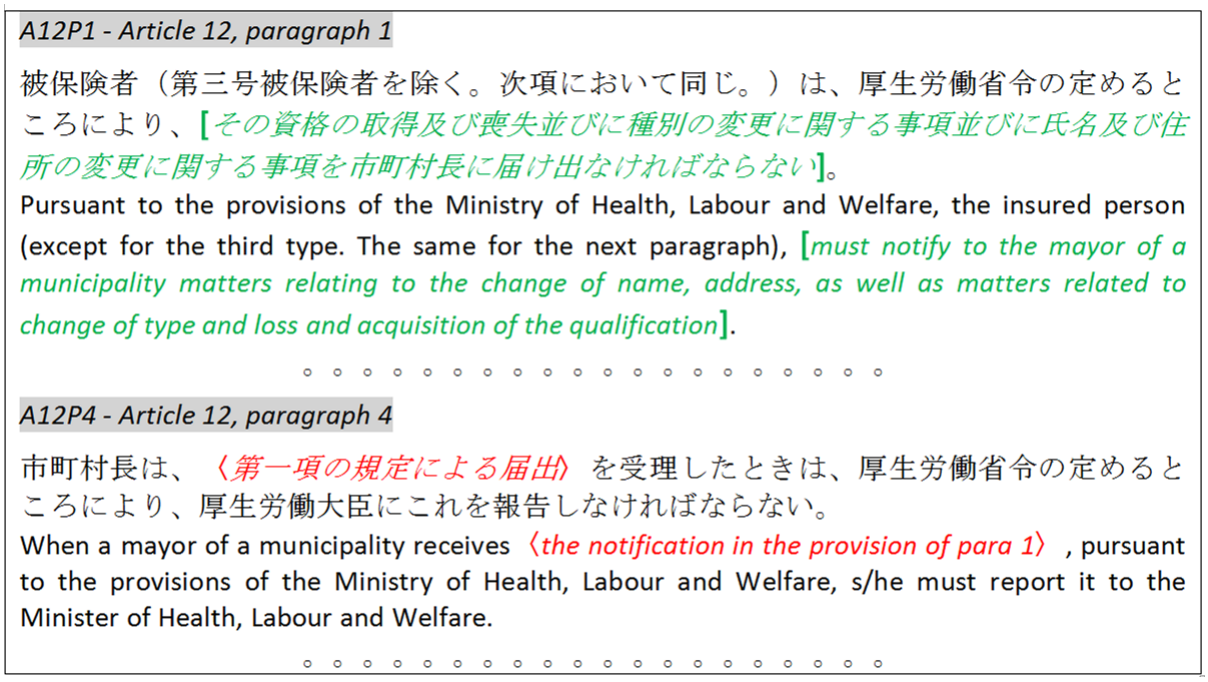

Figure 1.2: An example of reference phenomena in legal texts. In this figure, references are bounded by red angle brackets (⟨⟩) while their referents are bounded by green square brackets ([ ]).

The reference phenomenon is not only popular in general texts but also in legal texts. At the discourse level, legal texts contain many reference phenomena. These references usually bring precious information. The law will be difficult to comprehend if we cannot read the referenced items within it. Resolving the reference phenomena, therefore, is

an important task in the Legal Engineering[50, 51] research. However, in comparison to general domains, little research has concentrated on reference resolution in legal texts. The main reasons are the complex and long legal sentences, specific terms, etc.

Figure 1.2 shows excerpts from two documents[1] named A12P1, and A12P4. These excerpts contains one reference (the red texts bounded by red angle brackets), i.e. '*the notification in the provision of para 1*' in the document A12P4. To comprehend the content of the document A12P4, it is important to know the referenced items. In other words, we need to know to which part of texts (the green texts bounded by green square brackets) this reference refers. This kind of references is very popular in legal documents because lawmakers usually import pieces of available information which have already been introduced in other documents by using briefer expressions. This, as a result, helps to guarantee the soundness as well as the consistency in a law system. We name these briefer expressions '*references*', and their referenced items '*referents*'.

Previous work in this field mostly focuses on detecting and resolving so-called normative references to distinguish them from the above references. Normative references are slightly different from the above references. In the above examples, normative references would appear in the forms of '*para 1*'. In resolving these normative references, authors limit resolvers to identify only the referred documents but not to which parts of texts in these documents. With this output, users/lawmakers need to read over the referenced document to find which part of texts is actually referred to. This is somewhat redundant because that document may contain unnecessary information for the comprehension of the input sentence.

To avoid over-reading these unnecessary texts in the referenced document, in this thesis, we go a step further. Our reference resolver tries to extract the smallest fragments of texts that are actually referred to by references (the texts in green square brackets). Resolving this type of references is more difficult because it requires syntactic and semantic understanding of references and their context information as well as of the referenced document that contains the referenced texts. In particular, in the example sentence in Figure 1.2, we extract the full phrase and resolve it to the smallest fragment of texts that describes the type of the notification in Paragraph 1, i.e. '*notification of matters relating to the change of name, address, as well as matters relating to change of type and loss and acquisition of the qualification*'.

Reference information has many benefits not only in supporting the understanding of texts, but also in the development of a better performance for many high-level tasks in NLP. Instead of studying applications of reference resolution in general texts, which have been implemented in much previous work, in this thesis, we investigate an application of reference resolution to a question answering (QA) system restricted to legal documents. In the legal domain, QAs could be applied to help citizens and lawmakers more easily access legal information. Previous work [2, 28, 85, 111] showed that a common problem is that traditional QAs are not adequate to find the correct answers to legal questions. Until now, however, there has been no research on QA using this advantage of references to help finding the correct answers. Much works dedicated to QAs in the legal domains [2, 28, 85, 111] has mostly focused on legal questions whose answers can be found in only one document. However, the fact is that there exist many legal questions requiring answers that combined from two documents which are linked based on references. This

[1]The term '*documents*' corresponds to articles, paragraphs, items, or sub-items according to the naming rules used in the legal domain.

type of questions is not adequately considered in previous research.

1.2 Focus of Research

Figure 1.3 illustrates the overall framework of this thesis. In this research, we focus on solving the reference resolution task in general texts and also in a restricted domain - the legal domain. Moreover, our research also aims at analyzing the effects of applying reference resolution to a QA system in the legal domain. The main contributions of our thesis are listed as follows:

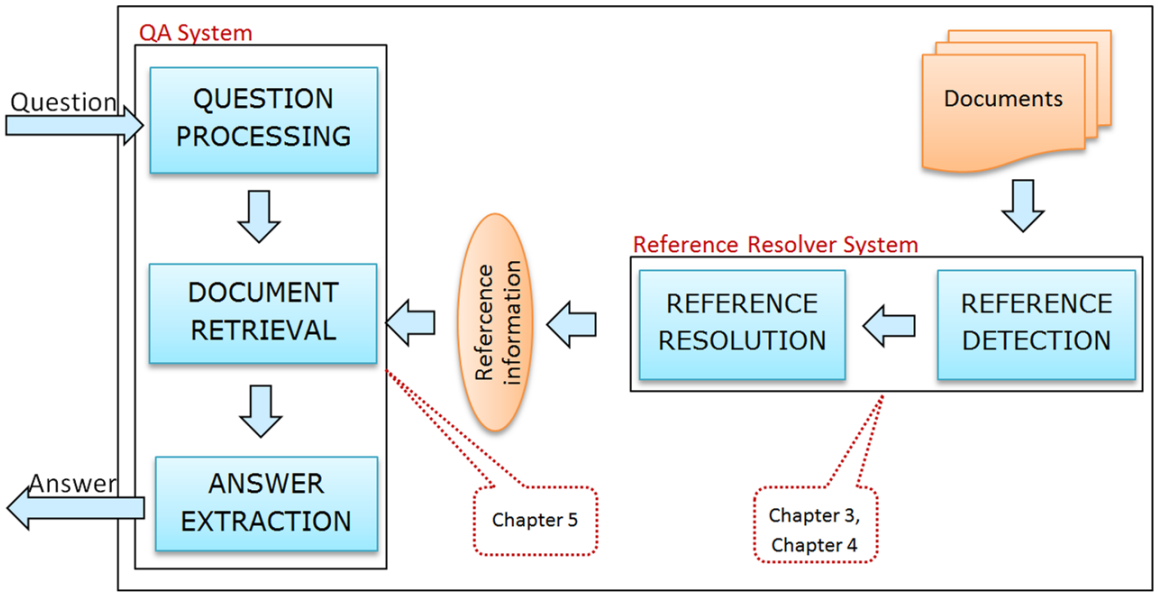

Figure 1.3: An overall framework of the thesis.

- **Coreference resolution in general texts:** In this research, we present an empirical study on a listwise approach to the CoRe task. This approach exploits a listwise learning-to-rank method which considers all antecedent candidates simultaneously, not only in the resolution phase but also in the training phase. In the training phase, a listwise algorithm is selected to train a co-reference resolution model which minimizes a listwise loss function and captures the ranking problems more naturally. In the resolution phase, the model assigns each candidate with a score that expresses the degree to which the candidate is co-referent with a given mention. Experimental results on the corpora of SemEval-2010 shared task 1 (the task of Co-reference resolution in multiple languages) show that our proposed system yields a good performance in multiple languages when compared to previous participating systems as well as a baseline pairwise system using the ranking support vector machine as the learning algorithm.

- **Reference resolution in legal texts:** This thesis also investigates the task of reference resolution in the legal domain. The aim is to create a system which can automatically extract referents for references in real time. This is a new interesting task in Legal Engineering research. Based on the structures of references in legal

texts, we propose a four-step framework to accomplish the task. We also show how machine learning approaches can be exploited on each step rather than using previous rule-based approaches. The final system achieves 80.06% in the F1 score for detecting references, 85.61% accuracy for resolving them, and 67.02% in the F1 score on the end-to-end setting task on the Japanese National Pension Law corpus.

- **Question Answering in legal texts:** Finally, in this thesis, we investigate an application of reference resolution to a QA system restricted to legal documents. We focus on one type of questions which can be of much benefit from the reference information. Based on the characteristics of the law sentences and the reference information between them, we propose a five-step framework to help extracting the answer to this type of question. Experimental results show that the proposed method is quite effective and outperforms a baseline, which does not utilize reference information.

1.3 Thesis Outline

This thesis consists of six chapters. The thesis is structured as follows:

Chapter 2 - In this chapter, we present some statistical machine learning methods used in this thesis. In the first section, we describe sequence labeling problems and then present a typical and effective algorithm to perform the task, i.e. Conditional Random Fields [54] (CRFs). Next, we introduce two strong classifiers to perform classification tasks, namely Maximum Entropy Models [97] (MEMs) and Support Vector Machines [21, 115] (SVMs). Finally, we describe in this thesis the task of learning-to-rank [57, 59] applied in candidate rankings of several sub-tasks.

Chapter 3 - This chapter presents an empirical study on a listwise approach to coreference resolution. This method allows us to consider all antecedent candidates simultaneously not only in the training phase but also in the resolution phase. First, we review traditional models which were previously proposed for the coreference resolution task. Then, we describe a listwise approach to this task. We begin this section by motivating the use of a ranker for coreference resolution. After that, we present the learning-to-rank task as well as two common and effective listwise approaches. We also show how to model this listwise approach for the coreference resolution task. Next, we describe experimental setups and the performance of the proposed listwise approach in comparison to previous approaches.

Chapter 4 - This chapter presents a study on resolving references in legal texts. First, we review some related work. Then, we describe some characteristics of references in legal texts. Based on these characteristics, we propose a four-step framework to solve this task. We also present solutions for each step in the proposed framework. Finally, we describe experiments. In this section, we also analyze the impact of each step on the whole system and illustrate an output example of the final system. In addition, we propose a semi-supervised technique to improve the performance of the final system.

Chapter 5 - This chapter presents a study on exploiting reference information to build a QA system restricted to the legal domain. We focus on answering a type of questions whose answers cannot be extracted from merely one document. To the best of our knowledge, this type of questions is not adequately considered in previous research. To cope with these, we propose a novel approach which allows exploiting the reference

information between legal documents to find answers to this type of legal questions. This approach also uses the requisite-effectuation structures of legal sentences and some effective similarity measures based on legal terms to support finding correct answers without training data.

Chapter 6 - In this final chapter, we first summarize the three main tasks of our thesis including the main achievement and contributions, as well as remaining problems. Next, we consider possible future research direction by mentioning open problems that would be interesting to address.

Chapter 2

Background: Backgrounds on Statistical Machine Learning Models Applied in NLP

2.1 Sequence Labelling

The need to segment and label sequences arises in many different problems in several scientific fields, especially in natural language processing (NLP) (i.e. named entity recognition [31, 118], POS tagging [113, 124], text chunking [56], etc.). There are many models proposed to solve this problem such as Hidden Markov Model (HMM) [9, 93], maximum entropy Markov models (MEMMs) [11, 67, 112], etc. Among them, Conditional random fields (CRFs) [54, 107, 108] offer several advantages over HMMs and stochastic grammars for such tasks, including the ability to relax strong independence assumptions made in those models. CRFs also avoid a fundamental limitation of MEMMs and other discriminative Markov models based on directed graphical models, which can be biased towards states with few successor states. CRFs outperform both MEMMs and HMMs on a number of real-world sequence labeling tasks [54, 89, 103].

CRFs are a class of statistical modeling method often applied in pattern recognition and machine learning, where they are used for structured prediction. A CRF can take context into account, whereas an ordinary classifier predicts a label for a single sample without regard to 'neighboring' samples. The linear chain CRF popular in natural language processing predicts sequences of labels for sequences of input samples.

Lafferty et al. [54] define a CRF on observation X and random variable Y as follows:

Definition: Let $G = \langle V, E\rangle$ be a graph such that $Y = (Y_v)_{v\in V}$, so that Y is indexed by the vertices of G. Then (X, Y) is a conditional random field in case, when conditioned on X, the random variables Y_v obey the Markov property with respect to the graph: $p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, w \sim v)$, where $w \sim v$ means that w and v are neighbors in G.

What this means is that a CRF is an undirected graphical model whose nodes can be divided into exactly two disjoint sets X and Y, the observed and output variables, respectively; the conditional distribution $p(Y|X)$ is then modeled. Figure 2.1 illustrates the simplest and most common graph structure in which the nodes corresponding to elements of Y from a simple first-order chain. The probability of a label sequence y given an observation sequence x can be written as:

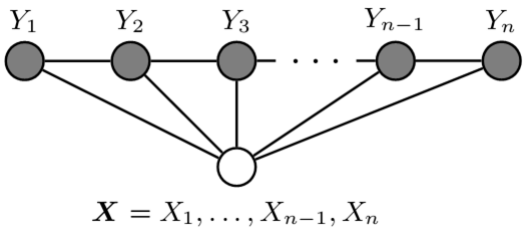

Figure 2.1: Graphical structure of a chain-structured CRFs for sequences.

$$p(y|x,\lambda) = \frac{1}{Z(x)} exp(\sum_j \lambda_j F_j(y,x)) \tag{2.1}$$

where $Z(x)$ is a normalization factor, and

$$F_j(y,x) = \sum_{i=1}^{n} f_j(y_{i-1}, y_i, x, i)$$

where $f_j(y_{i-1}, y_i, x, i)$ is either a state function or a transition function.

Assuming the training data $\{(x^{(k)}, y^{(k)})\}$ are independenly and identically distributed, the product of (2.1) overall training sequences, as the function of the parameters λ, is known as the likelihood. Maximum likelihood training chooses parameter values such that the logarithm of the likelihood, known as the log-likelihood, is maximized. For a CRF, the log-likelihood is given by:

$$L(\lambda) = \sum_k \left[log \frac{1}{Z(x^{(k)})} + \sum_j \lambda_j F_j(y^{(k)}, x^{(k)}) \right].$$

This function is concave, guaranteeing convergence to the global maximum.

Differentiating the log-likelihood with respect to parameter λ_j gives:

$$\frac{\partial L(\lambda)}{\partial \lambda_j} = E_{\tilde{p}(Y,X)} \left[F_j(Y,X) \right] - \sum_k E_{p(Y|x^k,\lambda)} \left[F_j(Y, x^{(k)}) \right],$$

where $\tilde{p}(Y,X)$ is the empirical distribution of training data and $E_p[.]$ denotes expectation with respect to distribution p. Note that setting this derivative to zero yields the maximum entropy model constraint: The expectation of each feature with respect to the model distribution is equal to the expected value under the empirical distribution of the training data.

It is not possible to analytically determine the parameter values that maximize the log-likelihood setting the gradient to zero and solving for λ does not always yield a closed form solution. Instead, maximum likelihood parameters must be identified using an iterative technique such as iterative scaling [24, 88] or gradient-based methods [103, 117].

2.2 Some Robust Classifiers

Many problems in NLP can be viewed as linguistic classification problems, in which linguistic contexts are used to predict linguistic classes. This section presents two robust

classifiers, i.e. Maximum Entropy Models (MEMs) and Support Vector Machines (SVMs), which are successfully used in many applications in NLP such as morphological analysis, text chunking, named entity recognition, etc.

2.2.1 Maximum Entropy Models

Maximum Entropy Models (MEM) [97] are a method of estimating the conditional probability $p(y|x)$ that a model outputs a label y given a context x:

$$p(y|x) = \frac{1}{Z(x)} exp(\sum_i \lambda_i f_i(x, y))$$

where $f_i(x, y)$ refers to a feature function; λ_i is a parameter of the model; and $Z(x)$ is a normalization factor. For example, in part-of-speech (POS) tagging problem, y is a POS tag and x is the context of a word (the word itself and its surrounding words) in a sentence. To capture statistic information, this method requires that the model accord with some constraints which have the form:

$$p(f) = \tilde{p}(f).$$

In this formula, f is a feature function (or feature for short), which takes a pair (x, y) as input and outputs a real value. Usually, f is a binary-value indicator function. For example, in POS tagging task, a feature function can be expressed as follows:

$$f(x, y) = \begin{cases} 1 & \text{if } y = Noun \text{ and the current word in x is } book, \\ 0 & \text{otherwise.} \end{cases}$$

$p(f)$ and $\tilde{p}(f)$ are the expected values of f with respect to the model $p(y|x)$ and the empirical distribution $\tilde{p}(x, y)$, respectively. They are defined as follows:

$$p(f) \equiv \sum_{x,y} \tilde{p}(x)p(y|x)f(x, y),$$

$$\tilde{p}(f) \equiv \sum_{x,y} \tilde{p}(x, y)f(x, y),$$

where $\tilde{p}(x)$ is the empirical distribution of x in the training samples.

Suppose that we have n feature functions $f_i(i = 1, 2, \ldots, n)$ and want our model to accord with these statistics. Our model will belong to a subset Q of P (the set of all conditional probability distributions) defined by

$$Q \equiv \{p \in P | p(f_i) = \tilde{p}(f_i), i = 1, 2, \ldots, n\}.$$

The maximum entropy method chooses the model $p^* \in Q$ that maximizes the entropy function $H(p)$:

$$p^* = \text{argmax}_{p \in Q} H(p)$$

where the entropy function $H(p)$ is defined as follows:

$$H(p) \equiv -\sum_{x,y} \tilde{p}(x)p(y|x) \log p(y|x).$$

To solve the constrained optimization problem, we first convert the primal problem to a dual optimization problem using the method of Lagrange multipliers [11]. Then the solution of the dual optimization problem can be found by applying the improved iterative scaling method [11, 24] or LBFGS method [79].

Maximum entropy model has been applied successfully to many NLP task including POS tagging [97], statistical machine translation [11, 29], etc.

2.2.2 Support Vector Machines

Support Vector Machines (SVMs) [16, 17, 18, 21] is a statistical machine learning technique proposed by Vapnik et al. It is not only well motivated in the theoretical aspect, but also yields good performance in the empirical aspect (including computer vision, handwriting recognition, pattern recognition, and statistical natural language processing). Let's take the simplest case to study on how SVMs work. This is called 2-class classification: $x \in R^n$ is some objects and $y \in \{-1, 1\}$ is a class label. SVMs choose a hyperplane separating samples in a classification task. In the field of natural language processing, SVMs have been applied to text categorization, word sense disambiguation, text chunking, syntactic parsing, semantic parsing, discourse parsing, etc., and achieved very good results.

In linear case, we assume that we want to find a hyperplane that separates positive and negative samples. Suppose that we have n training samples:

$$\{(x_i, y_i)\}_{i=1}^{n}, x_i \in R^m, y_i \in \{+1, -1\},$$

where x_i is the feature vector and y_i is the class (or label) of the i^th sample.

The goal is to separate the positive and negative samples by a hyperplane in the form:

$$w.x + b = 0,$$

where $w \in R^m$ and $b \in R$ are parameters.

Among the set of all possible hyperplanes, SVMs will find an optimal hyperplane (correspond to find an optimal parameter set for w and b). In the SVMs framework, the optimal hyperplane is the hyperplane with maximal **margin** between two classes. Figure 2.2 illustrates this strategy. Solid lines show two possible hyperplanes (or candidates). Each candidate separates correctly the training samples into two classes. Two dashed lines parallel to the candidate indicate the boundaries in which the candidate can be moved without any misclassication. The distance between those parallel dashed lines is called by **margin**.

Suppose that the training samples satisfy the following constraints:

$$w.x_i + b \geq +1 for\ y_i = +1$$

$$w.x_i + b \leq -1 for\ y_i = -1$$

These constraints can be combined into the following inequalities:

$$y_i(w.x_i + b) - 1 \geq 0,\ \forall i$$

Figure 2.3 shows how to calculate the margin. We have, the perpendicular distance from the origin to the solid line $w.x + b = 0$ is $\frac{|b|}{||w||}$, where $||w||$ is the Euclidean norm of w.

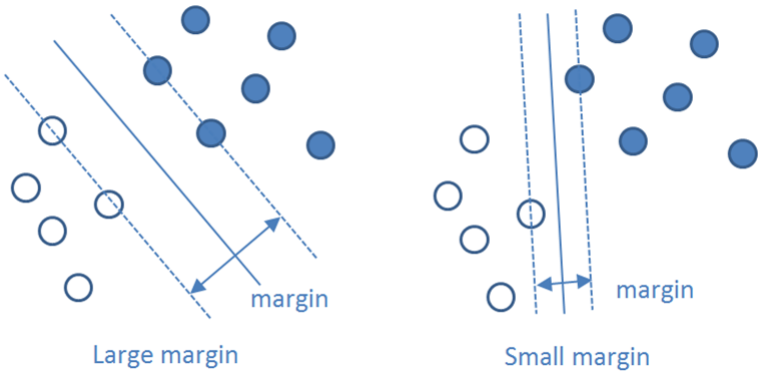

Figure 2.2: Small margin and large margin.

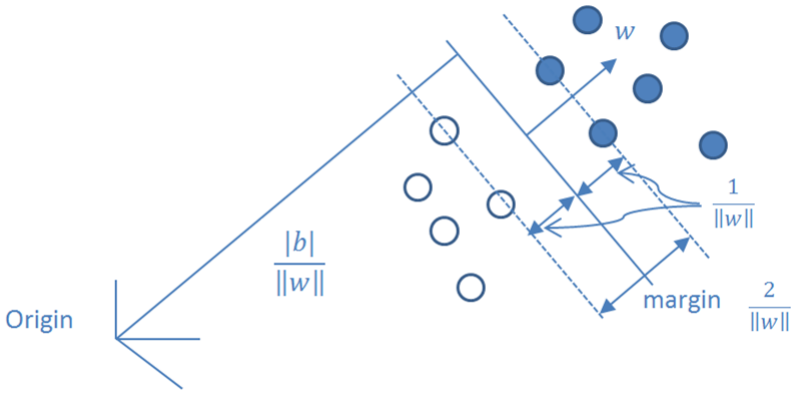

Figure 2.3: Calculation of the margin in SVMs framework.

Similarly, the perpendicular distances from the origin to two dashed lines ($w.x+b=1$ and $w.x+b=-1$ are $\frac{|b-1|}{||w||}$ and $\frac{|b+1|}{||w||}$. Let d_+ and d_- be the distances between the solid lines and two dashed lines. We will have $d_+ = d_- = \frac{1}{||w||}$. Hence, the margin $M = d_+ + d_- = \frac{2}{||w||}$.

To maximize the margin M, we minimize $||w||$. The task now becomes solving the following optimization problem:

Minimize:

$$L(w) = \frac{1}{2}||w||^2$$

Subject to:

$$y_i(w.x_i + b) - 1 \geq 0, \forall i = 1, 2, ..., n.$$

The training samples which lie on two dashed lines are called support vectors. In the cases where we cannot separate training samples linearly (because of some noise in the training data, for example) we can build the separating hyperplane by allowing some misclassications. In those cases, we can build an optimal hyperplane by introducing a soft margin parameter, which trades off between the training error and the magnitude of the margin.

SVMs also can deal with non-linear classification problems. First, the optimization problem is rewritten into a dual form, in which feature vectors only appear in the form

of their dot products. By introducing a kernel function $K(x_i, x_j)$ to substitute the dot product of x_i and x_j in the dual form, SVMs can solve non-linear cases.

2.3 Learning-to-Rank Methods

2.3.1 Introduction to Learning-to-Rank

Learning to rank [44, 57, 59, 94] is a type of supervised or semi-supervised machine learning problem, in which the goal is to automatically construct a ranking model from training data. This section focuses on this strong machine learning technique and its applications to the field of natural language processing (NLP). Specifically, we first introduce the ranking problem and distinguish it from other popular tasks such as classification, regression, and ordinal classification. Second, we present three major approaches to learning to rank which are the pointwise approach, the pairwise approach, and the listwise approach.

Learning to rank has been recently emerged in the past decade. Its purpose is to rank, i.e. produce a permutation of items in new, unseen lists in a way, which is 'similar' to rankings in the training data in some senses. Learning to rank algorithms have been applied in areas other than information retrieval, i.e. machine translation [119], recommender system [48], etc. To understand more about it, in this section, we would like to make a comparison to other traditional tasks such as classification and regression in the terms of the input, the output and the learning goals as follows:

Classification

The input is a feature vector $x \in R^d$, the output is a label $y \in Y$, and the goal is to learn a classifier $f(x)$ which can determine a class label y of a given feature vector x.

Regression

The input is a feature vector $x \in R^d$, the output is a real number $y \in R$, and the goal is to learn a function $f(x)$ which can determine a real number y of a given feature vector x.

Oridinal classification or ordinal regression

This is close to ranking, but is also different. The input is a feature vector $x \in R^d$, the output is a label $y \in Y$, representing a grade where Y is a set of grade labels. The goal of learning is to learn a model $f(x)$ which can determine the grade label y of a given feature vector x. The model first calculates the score $f(x)$, and then it decides the grade label y using a number of thresholds. Specifically, the model segments the real number axis into a number of intervals and assigns to each interval a grade. It then takes the grade of the interval which $f(x)$ falls into as the grade of x.

Learning to Rank

In ranking, one cares more about accurate ordering of objects, while in ordinal classification, one cares more about accurate ordered-categorization of objects. As will be seen later, ranking can be approximated by classification, regression, and ordinal classification.

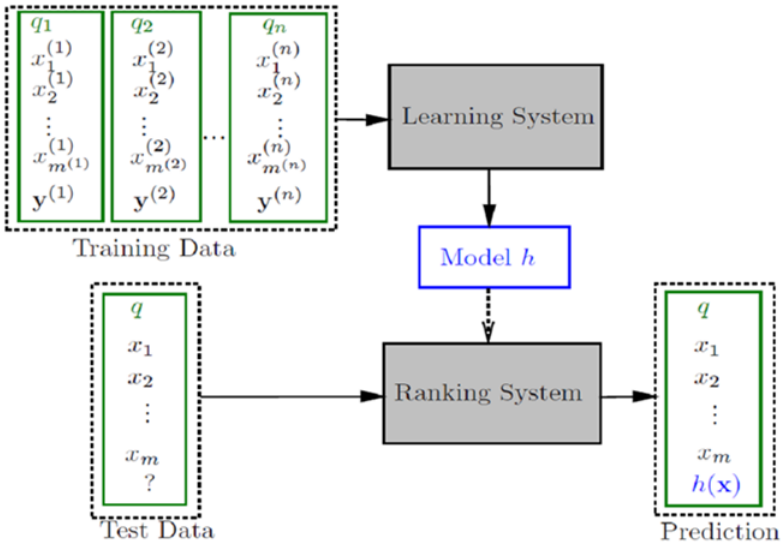

Figure 2.4: Learning-to-rank framework (cited from Liu [59]).

2.3.2 Major approaches in Learning-to-Rank

To give an overview of learning to rank, we choose information retrieval as an example as in [59]. Figure 2.4 shows the typical 'learning-to-rank' framework. The framework includes a training set which consists of n training queries $q_i(i = 1, 2, ..., n)$. Each query is associated with documents represented by feature vectors $x(i) = x_j^(i)(j = 1tom_i)$ where m_i is the number of documents associated with query q_i, and the corresponding relevance judgments. From this training data, a ranking model h is built by using a specific learning algorithm. This model is optimized so that the output of the model can predict the gold label in the training set as accurately as possible, in terms of a loss function. In the testing phase, the model h will be used to sort documents of a new query q and return the corresponding ranked list to the user as the response. According to Liu [59], existing algorithms for learning to rank problems can be categorized into three main groups by their input representation and the loss functions.

The pointwise approach

In this case it is assumed that each query-document pair in the training data has a numerical or ordinal score. Then learning-to-rank problem can be approximated by a regression problem given a single query-document pair, predict its score.

The pairwise approach

In this case learning-to-rank problem is approximated by a classification problem learning a binary classifier that can tell which document is better in a given pair of documents. The goal is to minimize average number of inversions in ranking.

The listwise approach

These algorithms try to directly optimize the value of one of the above evaluation measures, averaged over all queries in the training data. This is difficult because most evalu-

ation measures are not continuous functions with respect to ranking model's parameters, and so continuous approximations or bounds on evaluation measures have to be used.

2.3.3 Algorithms for Learning-to-Rank

In this section, we will present some representative algorithms for each approach above. For the poitwise approach, we present the one-class SVM. For the pairwise approach, we present the Ranking SVM algorithm. For the listwise approach, we present two common algorithms which are ListNet and ListMLE. These algorithms will be used later in this thesis.

One-Class Support Vector Machine - OCSVMs

We have already presented the SVMs algorithm in the previous section.

Ranking SVMs

This is one of the first learning to rank methods, proposed by Herbrich et al. [41]. The idea is to transform ranking into pairwise classification and employ the SVM technique [21] to perform the learning task.

Assume that $X \in R^d$ is the feature space and $x \in X$ is an element in the space (feature vector). Further suppose that f is a scoring function $f : X \longrightarrow R$. Then one can rank feature vectors (objects) in X with $f(x)$. That is to say, given any two feature vectors $x_i, x_j \in X$, if $f(x_i) > f(x_j)$, then x_i should be ranked ahead of x_j, and vice versa.

$f(x)$ can be arbitrary, however, to simplify we suppose that $f(x)$ is a linear function in that:

$$f(x) = \langle w, x \rangle ,$$

where w denotes a weight vector and $\langle . \rangle$ denotes inner product.

We can transform the ranking problem into a binary classification problem if the scoring function is a linear function because of the reasons as follows:

- First, the following relation holds for any two feature vectors x_i and x_j, when $f(x)$ is a linear function.
$$f(x_i) > f(x_j) \leftrightarrow \langle w, x_i - x_j \rangle > 0.$$
- Next, for any two feature vectors x_i and x_j, we can consider a binary classification problem on the difference of the feature vectors $x_i - x_j$. Specifically, we assign a label y to it.

$$y = \begin{cases} +1 & \text{if } x_i \text{ - } x_j > 0 \\ -1 & \text{if } x_i \text{ - } x_j < 0 \end{cases}$$

Hence, $\langle w, x_i - x_j \rangle > 0 \leftrightarrow y = +1$

Therefore, the following relation holds. That is to say, if x_i is ranked ahead of x_j, then y is $+1$, otherwise, y is -1.

$$x_i > x_j \leftrightarrow y = +1$$

RankingSVM applies the SVM technology to perform pairwise classification. Given n training queries $\{q_i\}_{i=1}^{n}$, their associated document pairs $x_u^{(i)}$, $x_v^{(i)}$ and the corresponding gold label $y_{u,v}^{(i)}$, the mathematical formulation of Ranking SVM is as shown below, where a linear scoring function is used $f(x) = w^T x$,

$$min\frac{1}{2}||w||^2 + \lambda \sum_{n}^{i=1} \sum_{u,v:y_{u,v}^{(i)}} \xi_{u,v}^{(i)}$$

$$s.t. w^T(x_u^{(i)} - x_v^{(i)}) \geq 1 - \xi_{u,v}^{(i)},\ if\ y_{u,v}^{(i)} = 1$$

$$\xi_{u,v}^{(i)} \geq 0, i = 1, ..., n$$

This objective function in Ranking SVM is very similar to that in SVM, where the term $\frac{1}{2}||w||^2$ controls the complexity of the model w. The difference with SVM lies in the constraints, which are constructed from document pairs. The loss function in RankingSVM is a hinge loss defined on document pairs. For example, for a training query q, if document x_u is labeled as being more relevant than document $x_v (y_{u,v} = 1)$, then if $w^T x_u$ is larger than $w^T x_v$ by a margin of 1, there is no loss. Otherwise, the loss will be $\xi_{u,v}$. Such a hinge loss is an upper bound of the pairwise $0-1$ loss.

This RankingSVM can inherit nice properties of SVM such as the ability to handle complete non-linear problems, etc.

ListNet

This sub-section describes the general setting for a learning-to-rank task. In a learning-to-rank task, a set of m samples $S = \{s^{(1)}, s^{(2)}, \ldots, s^{(m)}\}$ is given. Each sample $s^{(i)}$ consists of an object list $o^{(i)} = \{o_1^{(i)}, o_2^{(i)}, \ldots, o_{n^{(i)}}^{(i)}\}$, where $o_j^{(i)}$ denotes the j^{th} object and $n^{(i)}$ denotes the number of objects in i^{th} sample. Furthermore, each object list $o^{(i)}$ is associated with a list of scores $y^{(i)} = \{y_1^{(i)}, y_2^{(i)}, \ldots, y_{n^{(i)}}^{(i)}\}$, where $y_j^{(i)}$, a real number, is the score of the object $o_j^{(i)}$. In coreference resolution task, a sample $s^{(i)}$ is associated with a mention $m^{(i)}$to be resolved, each object $o_j^{(i)}$ corresponds to an antecedent candidate $c_j^{(i)}$, and score $y_j^{(i)}$ denotes the judgment on an antecedent candidate $c_j^{(i)}$ with respect to the mention $m^{(i)}$ (the value of $y_j^{(i)}$ expresses how relevant coreference an antecedent candidate $c_j^{(i)}$ is with a mention $m^{(i)}$ to be resolved).

A feature function ϕ will produce a real-value feature vector for each object $x_j^{(i)} = \phi(o_j^{(i)})$ $(i = 1, 2, \ldots, m; j = 1, 2, \ldots, n^{(i)})$. A list of feature vectors $x^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \ldots, x_{n^{(i)}}^{(i)}\}$ and the corresponding list of scores $y^{(i)} = \{y_1^{(i)}, y_2^{(i)}, \ldots, y_{n^{(i)}}^{(i)}\}$ will form a training instance $(x^{(i)}, y^{(i)})$. The training set can be represented by the following set: $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$.

In training phase, we want to learn a ranking function f, that produces a real-valued score $f(x_j^{(i)})$ for each feature vector $x_j^{(i)}$.

Suppose that $z^{(i)} = \left(f(x_1^{(i)}), f(x_2^{(i)}), \ldots, f(x_{n^{(i)}}^{(i)})\right)$ is the list of scores produced by f on a list of feature vectors $x^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \ldots, x_{n^{(i)}}^{(i)}\}$, and L is a loss function defined on two lists of scores $y^{(i)}$ and $z^{(i)}$. We want to minimize the total losses on the training data.

In ranking phase, given a new sample s' (a list of new objects o'), we first construct a list of feature vectors x' using feature function ϕ, and then produce a list of scores y' using ranking function f. Finally, objects are ranked in descending order of the scores.

Next, we will give a brief introduction to two listwise learning-to-rank methods, i.e ListNet [19] and ListMLE [122] (next sub-section). ListNet is a listwise method for learning-to-rank, which uses Cross Entropy metric as loss function, Neural Network as model, and Gradient Descent as learning algorithm. If we use a linear Neural Network model, the score of a feature vector can be calculated as follows:

$$f_\omega(x_j^{(i)}) = \left\langle \omega, x_j^{(i)} \right\rangle$$

where $\langle ., . \rangle$ denotes an inner product.

The learning algorithm of ListNet method is presented as Algorithm 1, where T is the number of iterations and η is the learning rate. The core of the algorithm consists of two main steps doing on each training sample:

1. Compute the score list for the sample using the current value of parameter vector ω.

2. Update the parameter vector ω using gradient $\Delta\omega$.

Algorithm 1 Learning Algorithm of ListNet method (cited from Cao et al. 2007)

Input: Set of training instances: $(x^{(i)}, y^{(i)})_{i=1}^{m}$
Parameter: iteration number T and learning rate η
Initialize parameter ω
for $t = 1 \to T$ **do**
  for $i = 1 \to m$ **do**
    Input $x^{(i)}$ to Neural Network and Compute score list $z^{(i)}(f_\omega)$ with current value of ω
    $z^{(i)}(f_\omega) = \left(f_\omega(x_1^{(i)}), \ldots, f_\omega(x_{n^{(i)}}^{(i)})\right)$
    Compute gradient $\Delta\omega$ using equation (2.2)
    Update $\omega = \omega - \eta \times \Delta\omega$
  end for
end for
Output: Neural Network model ω

The gradient $\Delta\omega$ is computed using the loss function L as follows[1]:

$$\begin{aligned}
\Delta\omega &= \frac{\partial L(y^{(i)}, z^{(i)}(f_\omega))}{\partial\omega} \\
&= -\frac{1}{\sum_{j=1}^{n^{(i)}} exp(y_j^{(i)})} \sum_{j=1}^{n^{(i)}} exp(y_j^{(i)}) \frac{\partial f_\omega(x_j^{(i)})}{\partial\omega} \\
&+ \frac{1}{\sum_{j=1}^{n^{(i)}} exp(f_\omega(x_j^{(i)}))} \sum_{j=1}^{n^{(i)}} exp(f_\omega(x_j^{(i)})) \frac{\partial f_\omega(x_j^{(i)})}{\partial\omega}
\end{aligned} \tag{2.2}$$

[1]In Algorithm 1, it is not necessary to compute the loss function L. The gradient $\Delta\omega$ can be computed directly based on the derivation of L (see Cao et al., [19] for more details).

ListMLE

ListMLE [122] is a listwise method which uses the likelihood loss as the loss function. Like ListNet method, it also uses Neural Network as model. The score of a feature vector is computed in the similar way:

$$f_\omega(x_j^{(i)}) = \left\langle \omega, x_j^{(i)} \right\rangle$$

The learning algorithm of ListMLE method is presented as Algorithm 2, where ϵ is the tolerance rate and η is the learning rate. The core of the ListMLE algorithm also consists of two main steps which compute the score list for the sample using the current value of parameter vector ω, and then update the parameter vector ω using gradient $\Delta\omega$.

Algorithm 2 Learning Algorithm of ListMLE method (cited from Xia et al. 2008)

Input: Set of training instances: $(x^{(i)}, y^{(i)})_{i=1}^m$
Parameter: Tolerance rate ϵ and learning rate η
Initialize parameter ω
  repeat
    for $i = 1 \to m$ **do**
      Input $x^{(i)}$ to Neural Network and compute score list $z^{(i)}(f_\omega)$ with current value of ω
      $z^{(i)}(f_\omega) = \left(f_\omega(x_1^{(i)}), \ldots, f_\omega(x_{n^{(i)}}^{(i)})\right)$
      Compute gradient $\Delta\omega$ using equation (2.3)
      Update $\omega = \omega - \eta \times \Delta\omega$
    end for
    Compute likelihood loss L using equation (2.4)
  until change of likelihood loss is below ϵ
Output: Neural Network model ω

In ListMLE method, the gradient $\Delta\omega$ is computed using the following formula:

$$\Delta\omega = \frac{\sum_{t=1}^{n(i)} x_{\pi^{-1}(t)}^{(i)} exp(f_\omega(x_{(\pi^i)^{-1}(t)}^{(i)}))}{\sum_{t=1}^{n(i)} exp\left(f_\omega(x_{(\pi^i)^{-1}(t)}^{(i)})\right)} - x_{(\pi^{(i)})^{-1}(1)}^{(i)} \tag{2.3}$$

and likelihood loss L is computed using the following formula:

$$L = -\sum_{i=1}^{m} log \frac{exp\left(f_\omega(x_{(\pi^{(i)})^{-1}(1)}^{(i)})\right)}{\sum_{t=1}^{n(i)} exp\left(f_\omega(x_{(\pi^{(i)})^{-1}(t)}^{(i)})\right)} \tag{2.4}$$

In that: $\pi^{(i)}$ is the perfect (gold) ranking by $y^{(i)}$, $\pi^{(i)}(k)$ returns the ranking place of the k^{th} element, $(\pi^{(i)})^{-1}$ is the inverted mapping of $(\pi^{(i)})$, and $(\pi^{(i)})^{-1}(t)$ returns the position of the element which is ranked at the t^{th} place.

Chapter 3

An Empirical Study on a Listwise Approach to Coreference Resolution using Learning-to-rank

To realize coreference resolution, this chapter presents a listwise approach, which exploits a listwise learning-to-rank method. This approach allows to consider all antecedent candidates simultaneously not only in the resolution phase but also in the training phase.

3.1 Introduction

Reference resolution [47] (Chapter 21, Section 21.4) is the task of determining which entities are referred to by which linguistic expressions. This task plays an important role in a large number of natural language processing (NLP) applications such as Text Sumarization [105], Question Answering [106], and Machine Translation [80, 84]. Therefore, it has attracted much attention within the NLP community. Among all types of reference phenomena, coreference is the most popular and is the focus of most researches on reference resolution. Many works on various aspects (such as linguistic features [37, 75], machine learning models [76, 104], multiple languages [99], and so on) of the coreference resolution task have been published.

To solve the coreference resolution task, a lot of models have been proposed. In mention-pair models [77, 91, 104], authors train a model to determine whether an antecedent candidate is coreferent with an anaphoric mention or not, and the antecedent will be chosen among candidates that are classified coreferent with an anaphoric mention. In entity-mention models [62, 126], authors consider a preceding cluster of mentions instead of single antecedent candidates. A model is trained to classify whether a mention and a preceding cluster are coreferent. These models suffer from an important weakness, which makes them unable to completely solve the problem. In these models, each candidate is resolved independently with the other candidates. Therefore, the probability assigned to each candidate merely encodes the likelihood of that particular candidate being coreferential with a given mention.

Mention-ranking models [25, 26, 74, 125] have been proposed to overcome the limitation of mention-pair models and entity-mention models. In this method, authors train a ranker which ranks candidates and the candidate with the highest rank will be chosen as the correct antecedent. The ranker can be trained using the limited memory variable

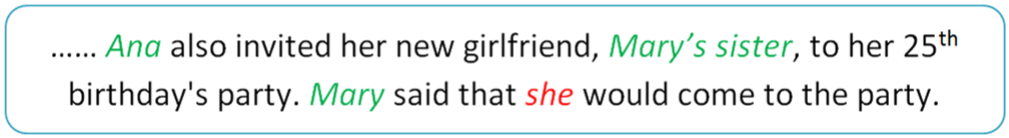

Figure 3.1: A motivating example of coreference resolution using a listwise approach.

metric algorithm [26] or using support vector machines [96]. Cluster-ranking model [95] is a method which combines the strength of entity-mention models and mention-ranking models. This method ranks the preceding clusters and chooses the best cluster which the anaphoric mention will belong to. Although mention-ranking models and cluster-ranking models allow all candidates to be evaluated together when deciding which candidate is the antecedent of a mention, they do not consider all candidates simultaneously when training the rankers. Most authors use tournament by [43] and twin-candidate model by [95, 125]. So, they only directly compare pairs of antecedent candidates by building a preference classifier based on the triple including two candidates and an anaphoric mention. This extension provides important benefits, however, the above weakness is not fully resolved.

Figure 3.1 shows an example dialog. In this dialog, if we only look at the context of one sentence '*Mary said that she would come to the party*', the most likely reading is that *she* refers to *Mary*. However, if we look at the broader context, *she* could instead refer to someone else (most likely someone introduced earlier in the dialog, i.e. *Ana* or *Mary's sister*). Therefore, a good reference resolver should have the ability to capture the whole context of each mention. In other words, it should estimate how good a candidate is in relation to other candidates, instead of only considering one or two candidates independently.

Moreover, by default, previous approach is strongly assumed that candidates or pairs of candidates are generated i.i.d[1] and the trained models will be biased towards mentions with more candidates. Another problem is that the objective of learning is formalized as minimizing errors in classification of candidates or pairs of candidates, rather than minimizing errors in ranking of candidates globally. Motivated from these drawbacks, we investigate a listwise approach, which is a more straightforward way to allow direct comparison of different candidate antecedents for an anaphoric mention. This idea used to be preliminarily referred to in the literature of anaphora resolution (i.e., Centering algorithm [14]). This is suitable with the fact that all antecedent candidates are closely related to each other in a given context of a document and should be considered simultaneously. In this study, we employ the listwise learning-to-rank method which is originally proposed for the learning-to-rank task in information retrieval [19], to solve the coreference resolution task. This listwise approach has been successfully applied to the information retrieval task [19] [122], question answering [1], etc. It has been shown to be more effective in comparison with other learning-to-rank methods such as pointwise and pairwise approaches which do not use lists of objects [2] as instances in learning [19].

Exploiting the listwise learning-to-rank method allows us to operate on the entire list of candidates in the training phase as well as the resolution phase. Specifically, lists of

[1]independent and identically distributed.

[2]In the coreference resolution task, objects means antecedent candidates; in information retrieval task, objects means document candidates; in question answering systems, objects means answer candidates.

candidates are used as 'instances' in learning. In the training phase, we train a ranking model which minimizes a listwise loss function and captures the ranking problem in a conceptually more natural way than previous ranking approaches [25, 26, 74, 125]. In the resolution phase, the model will assign each candidate antecedent with a score indicating how likely the candidate antecedent and the mention to be resolved are coreferent, and the candidate with the highest score will be selected as a correct antecedent.

In experiments, we implement two effective listwise algorithms which are ListNet[19] and ListMLE[122]. Experimental results on the corpora of SemEval-2010 shared task 1 (the task of *coreference resolution in multiple languages*) [99] show that when applied to coreference resolution, this listwise approach mostly yields better performance than previous approaches. In comparison to the best system SUCRE, we achieved comparative performance.

Our main contributions can be summarized in the following points:

1. Proposing a listwise approach to coreference resolution using listwise learning-to-rank methods.

2. Conducting experiments on multiple languages to show the performance of the listwise approach.

3. Investigating two common and effective listwise learning-to-rank methods and comparing listwise and pairwise approaches in the coreference resolution task.

The rest of this chapter is organized as follows. Section 3.2 presents traditional models which was previously proposed for coreference resolution. Section 3.3 describes our listwise approach to this task. We begin this section by motivating the use of a ranker for coreference resolution. We also show how to model this listwise approach to coreference resolution. Section 3.4 presents the corpora of this SemEval-2010 shared task 1, participating systems, and four evaluation metrics. In this section, we also present experimental results to compare the listwise approach to previous participating systems as well as a pairwise ranking baseline model. In addition, we report experimental results using two different methods of generating training instances. This section also illustrates P-R curves to give a more informative picture of the systems's performance. We also add some discussion about the listwise approach. Finally, Section 3.5 concludes the chapter.

3.2 Previous models for the coreference resolution task

In this section, we review previous models for the coreference resolution task including mention-pair models (Section 2.1), entity-mention models (Section 2.2), and mention-ranking and cluster-ranking models (Section 2.3). Among these models, mention-pair and entity-mention models only consider one candidate (or one cluster) at a time in both the resolution and the training phases. Both mention-ranking and cluster-ranking models only consider a pair of candidates (or a pair of clusters) in the training phase.

We begin this section by giving an example used to investigate coreference resolution models.

Captain Farragut was a good seaman, worthy of the frigate he commanded. His vessel and he were one. He was the soul of it.

In this example, we assume there are two entities which are the good seaman named Captain Farragut and his vessel. Each entity is referred by its own referring expressions - the former are blue texts and the later are red texts.

3.2.1 Mention-pair models

In the *mention-pair* models [77, 91, 104], the models have to build a classifier which can classify whether a candidate is coreferent with an anaphoric mention. Each training instance is created between the mention and each of its antecedent candidates. In this case, each candidate is considered independently of the others.

In the above example, if we consider the mention *it*, we have to determine the correct antecedent among its candidates which are (*Captain Farragut*, *the frigate*, *he*, *His vessel*, *he* and *He*). This method generates three training instances correspondingly. Each training instance is represented by a feature vector built from each candidate and the anaphoric mention *it*. The built models will classify each of the pair between the mention *it* and its candidate antecedent are coreferent or not. The models's antecedent will be chosen among candidates that are classified coreferent with the mention by using the closest-first or the best-first strategies. In the closest-first strategy, the closest candidate that is classified as coreferent with the mention will be selected, while in the best-first strategy, the most probable preceding candidate that is classified as coreferent with the mention will be selected. If no such antecedent exists, the mention is considered as a non-anaphoric mention.

The mention-pair models have two weaknesses. First, the models only consider each candidate independently. So, they only determine how good a candidate is relative to the mention, but not how good a candidate is relative to other candidates. Second, they have limitations in their expressiveness. The information extracted from a candidate and the mention alone may not be enough for making a coreference decision [76].

3.2.2 Entity-mention models

In the *entity-mention models* [62, 126], the models classify whether a mention to be resolved and each of preceding clusters are coreferent or not. Training instances are created between the mention and a preceding cluster. These models improve the expressiveness problem by allowing the computation of the cluster-level features. They use cluster-level features which are computed from a feature employed by the mention-pair model by applying a logical predicate. Testing phase is like in *mention-pair models* except that we resolve the mention to the closest or the best preceding cluster that is classified as coreferent.

In the above example, if we consider the mention *it*, we have to determine the correct cluster to which *it* belongs. Here there are two clusters (*Captain Farragut*, *he*, *he* and *he*) and (*the frigate* and *his vessel*). Each referring expression in the cluster refers to the same real-world entity. In this case, the mention *it* belongs to the second cluster.

However, they still have their own weakness that is each cluster candidate is considered independently of the others.

3.2.3 Mention-ranking models and cluster-ranking models

In mention-pair and entity-mention models, each candidate for a mention to be resolved is estimated independently. Therefore, models cannot determine the most probable antecedent. In the later researches, there are also efforts to address this problem. Let's list out some works belonging to *mention-ranking models* [25, 26, 74, 125]. In mention-ranking models, they learn a ranker that can rank a set of candidates for each mention in a pairwise manner. Training examples build based on triple of two candidates and the anaphoric mention. This model has an additional constraint on the creation of instances: exactly only one of the two candidates can be coreferential with the anaphoric mention. In the resolution phase, the ranker will rank all the candidates (usually by assigning a score to each candidate), and the candidate with the highest rank will be chosen as a correct antecedent.

There is also another model which combines the strength of entity-mention model and mention-ranking model - the *cluster-ranking model* [95]. In that model, it ranks all preceding clusters for each mention to be resolved. Training examples are comprised of features between a mention to be resolved and its preceding cluster. The way of creating instances is same to the entity-mention model.

Although mention-ranking models and cluster-ranking models allow all candidates to be evaluated in a pairwise manner, they do not consider all candidates simultaneously especially when training the ranker. Therefore, the above weakness is not fully solved.

It should be noted that the number of training instances generated by these four models are quite large if we consider all mentions preceding an anaphoric mention as candidates. So, the model learned is easily biased to the anaphoric mention with a large number of training instances. Moreover, negative training instances can overwhelm positive training instances. To handle these problems, authors suggest using only a subset of preceding mentions as candidates.

3.3 A listwise approach to coreference resolution using learning-to-rank

In this section, we first discuss on why coreference resolution should be considered as a learning-to-rank problem. Then, we describe how to formulate coreference resolution in the view of a learning-to-rank problem. To solve this learning-to-rank task, we present two common and effective methods using listwise approaches which are ListNet [19] and ListMLE [122]. Lastly, we compare our model using a listwise learning-to-rank approach to previous models in terms of input, output, and loss function.

3.3.1 Coreference resolution as a learning-to-rank problem

References are frequently ambiguous and depend on the context. For example, in the sentence '*Mary said she would help me,*' *she* and *Mary* most likely refer to the same person or group, in which case they are coreferent. Though the most likely reading is that *she* refers to *Mary*, *she* could instead refer to someone else (most likely someone introduced earlier in a dialog). Hence, in this case, it is necessary to estimate how good the antecedent candidate *Mary* is in comparison to all available antecedent candidates of

the mention *she*. In other words, a coreference resolver should have the ability to examine all possible candidates at the same time. This is suitable with the fact that all antecedent candidates are closely related to each other in a given context of a document and should be considered together simultaneously. To perform this, learning-to-rank is a reasonable choice.

The idea that considers reference resolution as a ranking problem is actually presented in the literature of anaphora resolution [47]. A typical representative is Centering algorithm [14] in which when resolving anaphora, different analyses may correspond to different transition types that determine the final reference assignments. These transition types are ranked based on the criterion that an analysis involving least change is preferred. Another example is the method proposed by Lapping and Leass [55] in which the antecedent is selected on the basis of salience ranking and proximity.

In the aspect of the machine learning approaches, some authors performed the ranking idea for the co-reference resolution task such as the mention-ranking model [25, 26], cluster-ranking model [95], twin-candidates model [125], tournament model [43], etc. In these models, instead of only considering one candidate at a time, they consider two candidates at a time. They somehow address the drawback of previous models using classification approaches such as mention-pair or entity-mention models. However, these methods do not fully solve the problem in the aspect that it is only possible to utilize two candidates rather than the whole set of candidates in the training phase as well as the resolution phase. In other words, they forces different candidates for the same mention to be considered independently. This means that the strong independence assumptions hold during training phases. In fact, however, each candidate or candidate pair does not exist independently but it exists in a given context in relation with other candidates. Coreference resolution, therefore, is more appropriate to be resolved using all candidates instead of using only one or two candidates. To perform this, ranking approaches provides a more natural fit to the task than classification approaches. In this chapter, we investigate an approach using a listwise learning-to-rank method to solve the coreference resolution task. This listwise approach directly captures the competition among potential antecedent candidates, instead of considering each of them or pair of them independently. In training, it also learns a model using a ranking loss function rather than a classification loss function.

3.3.2 Formulating coreference resolution as a Learning-to-rank problem

We formulate the coreference resolution task as a learning-to-rank problem. This problem formulation has been used in Information Retrieval [19]. In the learning-to-rank framework, training data consists of a number of queries; each query is associated with a correctly-ranked list of document. For the coreference resolution task, each mention to be resolved plays the role as a query, and each of its antecedent candidates corresponds to a document in the information retrieval task. In the following section, we will describe the way of creating training instances, the training phase, and the resolution phase of this approach.

Formally, in training, a set of m samples $S = \{s^{(1)}, s^{(2)}, \ldots, s^{(m)}\}$ is given. Each sample $s^{(i)}$ is associated with a mention. Each $s^{(i)}$ consists of an antecedent candidate list $c^{(i)} = \{c_1^{(i)}, c_2^{(i)}, \ldots, c_{n^{(i)}}^{(i)}\}$, where $c_j^{(i)}$ denotes the j^{th} antecedent candidate and $n^{(i)}$ denotes the number of antecedent candidates for i^{th} sample. Furthermore, each antecedent

candidate list $c^{(i)}$ is associated with a list of scores $y^{(i)} = \{y_1^{(i)}, y_2^{(i)}, \ldots, y_{n^{(i)}}^{(i)}\}$, where $y_j^{(i)}$, a real number, is the score of the antecedent candidate $c_j^{(i)}$. In coreference resolution task, the score $y_j^{(i)}$ denotes the judgment on an antecedent candidate $c_j^{(i)}$ with respect to the mention $s^{(i)}$ (the value of $y_j^{(i)}$ expresses how relevant an antecedent candidate $c_j^{(i)}$ is coreferent with a mention $s^{(i)}$). Similar to the information retrieval task [3], in our task we define the score $y_j^{(i)}$ as one element in the set of $\{1, 0.5, 0\}$ with the meanings as follows:

$$y_j^{(i)} = \begin{cases} 0 & \text{if } c_j^{(i)} \text{ is not coreferent with mention } s^{(i)} \\ 1 & \text{if } c_j^{(i)} \text{ is coreferent with mention } s^{(i)} \text{ and closest to } s^{(i)} \\ 0.5 & \text{if } c_j^{(i)} \text{ is coreferent with mention } s^{(i)} \text{ and not closest to } s^{(i)} \end{cases}$$

In training, there maybe more than one antecedent candidate which is judged to be the correct antecedent of the current mention $s^{(i)}$. In this chapter, we propose a method for assigning scores to antecedent candidates using the coreferent criterion [76, 77] and the closest-first clustering strategy [104]. In our framework, we assign the highest score to the antecedent candidate which is coreferent with and is closest to the current mention $s^{(i)}$. The other candidates, which are coreferent with $s^{(i)}$, are assigned the lower scores. The remaining candidates, which are not coreferent with $s^{(i)}$, are assigned the score 0.

A feature function ϕ will produce a real-value feature vector $x_j^{(i)} = \phi(c_j^{(i)})$ for each antecedent candidate $c_j^{(i)}$ ($i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n^{(i)}$). A list of feature vectors $x^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \ldots, x_{n^{(i)}}^{(i)}\}$ and the corresponding list of scores $y^{(i)} = \{y_1^{(i)}, y_2^{(i)}, \ldots, y_{n^{(i)}}^{(i)}\}$ will form a training instance $(x^{(i)}, y^{(i)})$. The training set can be represented by the following set: $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$.

Training phase

In training phase, we want to learn a ranking function f, that produces a real-valued score $f(x_j^{(i)})$ for each feature vector $x_j^{(i)}$. With the usage of a linear Neural Network model, the score of a feature vector can be calculated as follows:

$$f_\omega(x_j^{(i)}) = \left\langle \omega, x_j^{(i)} \right\rangle,$$

where $\langle ., . \rangle$ denotes an inner product, ω is a vector of parameters of the model.

Suppose that $z^{(i)} = \left(f(x_1^{(i)}), f(x_2^{(i)}), \ldots, f(x_{n^{(i)}}^{(i)})\right)$ is the list of scores produced by f on a list of feature vectors $x^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \ldots, x_{n^{(i)}}^{(i)}\}$, and L is a loss function defined on two lists of scores $y^{(i)}$ and $z^{(i)}$. We want to minimize the total losses on the training data:

$$\sum_{i=1}^{m} L(y^{(i)}, z^{(i)})$$

From a score list, the listwise approach defines a probability distribution for a rank ordering over the list of candidates [19, 122]. In fact, each rank ordering corresponds to a

[3] In the information retrieval task, a score indicates the degree of relevance of a document to the corresponding query. It can be one element in the ordinal set, {perfect, excellent, good, fair, bad} or {5, 4, 3, 2, 1} in a numerical representation; or a score can also be a binary judgment in the set {relevant, not relevant} or {1, 0} in a numerical representation.

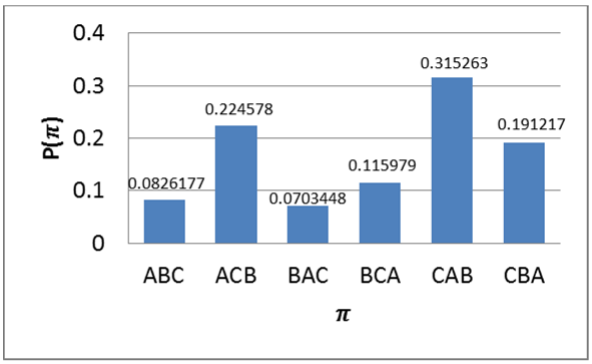

Figure 3.2: An example of a permutation probability distribution over three candidates named A, B, and C.

permutation of the list candidates. The probability of each permutation can be computed from its list of scores (in some ways). For example, the probability of permutation Π given the list of scores s can be defined as:

$$P_s(\Pi) = \prod_{j=1}^{n} \frac{\phi(s_{\Pi(j)})}{\sum_{k=j}^{n} \phi(s_{\Pi(k)})}$$

where n is the number of antecedent candidates, $\phi(.)$ is an increasing and strictly positive function, and $s_{\Pi(j)}$ denotes the score of the candidate at position j of permutation Π . In the ListNet method [19], the authors define $\phi(.)$ as an exponential function. An example of probability distribution defined by a ranking function f over three candidates named A,B, and C is given in Figure 3.2. In this example, the ranking function f assigns scores to candidates A, B, and C as 0.5, 0, and 1 respectively.

The loss function L measures the information loss on probability distributions calculated from the estimated scores $z^{(i)}$ and probability distributions calculated from the gold scores $y^{(i)}$. The purpose of the listwise approach is to minimize the total losses on the training data to learn the parameters of the model ω. With the use of top one probability, given two lists of scores we can view any metric between probability distributions as the listwise loss function. The ListNet method uses Cross Entropy as metric, then the listwise loss function becomes:

$$L(y^{(i)}, z^{(i)}) = -\sum_{j=1}^{n} P_{y^{(i)}}(j) log(P_{z^{(i)}}(j))$$

This listwise approach, therefore, allows us to examine all candidates at the same time.

Resolution phase

In ranking phase, given a new sample $s^{(')}$ (a list of new antecedent candidate $c^{(')}$), we first construct a list of feature vectors $x^{(')}$ using feature function ϕ, and then produce a list of scores $y^{(')}$ using ranking function f. Finally, objects are ranked in descending order of the scores. The candidate with a higher score will have higher probability to be coreferent with $s^{(')}$.

To determine the anaphoricity of a mention, usually a discourse status classifier is adopted to assist the identification of anaphoric mentions. This method requires the

Table 3.1: The main characteristics of all approaches.

	Mention-Pair Entity-Mention	**Mention-Ranking Cluster-Ranking**	**Listwise Approach (ListNet, ListMLE)**
Input	A single candidate presented by a feature vector x_i	A pair of candidates represented by feature vectors x_i, x_j	A set of candidates associated with a mention $X = (x_i)_{i=1}^{n}$
Output	classification y_i	pairwise classification $y_{i,j}$	Ranking list π_y
Model	classifier	classifier	Ranking model
Loss Function	Classification loss	Pairwise classification loss	Listwise ranking loss

understanding of characteristics of each language to determine which mention should be anaphoric. For example, [78] use a pool of feature sets including 37 features grouped in 6 types for a English anaphoricity determination system. They include lexical, grammatical (NP type), grammatical (NP property/relationship), Grammatical (Syntactic Pattern), Semantic and Positional feature types. These feature sets are not identical to all languages. This approach, therefore, is not convenient to implement for multiple languages. Another approach is the work of [95] in which they proposed an approach to joint discourse-new detection and coreference resolution for ranking model. This seems to be appropriate to conduct in multiple languages. However, experimental results (will be presented in Subsection 4.6) showed that this method did not yield better performance than the method that considers the ranker as an additional filter for detecting anaphora.

Like in mention-pair models, our model learns the degree that a candidate is coreferent with a given mention. Training instances for each mention also include negative and positive candidates. The only difference is that in training we optimize the parameters based on how good a candidate is in relation to remaining candidates, instead of considering each candidate independently. Therefore, information about the anaphoricity of a mention is also covered in the coreference relation between a mention and a list of its candidates. In our model, the output score is not the exact probability that a candidate is coreferent with a given mention. The score merely indicates the degree that a candidate is coreferent with a given mention in relation to other candidates. The higher the score of a candidate is, the higher the coreferent probability with a given mention is. In testing, to determine anaphoricity, we set up a threshold theta. If the given mention is non-anaphoric, it means that the score that it is coreferent with any candidate is below a threshold theta.

3.3.3 Comparing this listwise learning-to-rank model to previous models

Table 3.1 summarizes the main characteristics of the listwise approach in comparison with previous approaches in terms of input, output, loss function and model. With these characteristics, it should be noted that the first models (in the first column) consider only *one* antecedent candidate at a time; the second models (in the second column) consider only *two* antecedent candidates at a time. While, the proposed listwise approaches allow *all* antecedent candidates to be examined simultaneously.

Table 3.2: Parameter sets of ListNet and ListMLE algorithms

Model Name	Parameters	Languages	Values
ListNet	T - η - θ	English	2 - 0.005 - 0.12
		Catalan	2 - 0.001 - 0.15
		Spanish	5 - 0.01 - 0.25
		German	2 - 0.005 - 0.08
ListMLE	ϵ - η - θ	English	0.2 - 0.01 - 2
		Catalan	0.5 - 0.005 - 0.5
		Spanish	0.5 - 0.001 - 1.5
		German	0.5 - 0.005 - 0.36

3.4 Experiments and Results

This section presents our experiments on the coreference resolution task using our listwise approach. All experiments were conducted on the copora of the SemEval 2010 shared task 1 [99], which was organized to evaluate learning-based coreference resolution systems in multiple languages. To conduct experiments, we implemented ListNet [19] and ListMLE [122] methods by ourself. They are available for download now.

When using ListNet method, our system had to choose the set of three parameters which are (1) number of iteration T; (2) learning rate η; and (3) the threshold θ to determine a candidate is coreferent with a given mention or not. When using ListMLE method, our system had to choose the set of three parameters which are (1) tolerance rate ϵ; (2) learning rate η; and (3) the threshold θ to determine whether a candidate is coreferent with a given mention. To determine the best parameter set, we varied their values and selected parameters that yield the best coreference resolution system. Scoring coreference resolvers seems to be a continuing issue, with very little correlation between various methods. Each metric has its own advantages and disadvantages and there is no standard criterion which can estimate which one is better. In this research, therefore, we would like to use a popular method based summing up four metrics on F1 scores. Depending on real applications, if people wants to focus on a specific metric, they can set a higher weight to that metric and so on. The parameter sets for these two methods are listed in Table 3.2. After that, we used these parameters to evaluate our proposed systems on the test sets.

Next, we describe the SemEval-2010 shared task 1 (Sub-section 3.4.1), participating systems in the shared task (Sub-section 3.4.2), evaluation metrics (Sub-section 3.4.3), and feature sets which we used to learn coreference resolution models (Sub-section 3.4.4). Then, we present experimental results.

3.4.1 SemEval 2010 shared task on coreference resolution in multiple languages

Until the release of the SemEval-2010 shared task 1 [99], there has no competition or public corpus that allow us to evaluate different coreference resolution systems in multiple languages. Most published systems only focus on a specific language and use the same data sets such as ACE and MUC corpora to train and test the systems. This makes the systems easy to unintentionally adapt themselves to the corpus but not to the problem

in general. Therefore, the SemEval-2010 task 1 [99] made it possible to evaluate and compare various automatic coreference resolution systems in the following aspects:

- The portability of systems across languages.
- The relevance of different levels of linguistic information.
- The behavior of scoring metrics.

This shared task attracted lots of researchers' attentions, but finally only six teams submitted their final results. The participating systems differed in terms of architecture, machine learning methods, etc. These systems mostly based on the pairwise models. Unfortunately, these models suffered from an important weakness as discussed in the previous sections.

3.4.2 Participating systems

In this section, we preview previous approaches of the systems participating in the SemEval-2010 shared task which have the same experimental settings as in our experiments. The experimental results of those systems are also used to make an experimental comparison with our proposed approach's results. Here, we preview four systems: (1) RelaxCor system [102]; (2) SUCRE system [52]; (3) TANL-1 system [3]; and (4) UBIU system [128].

RelaxCor system

RelaxCor [102] is a constraint-based graph partitioning approach to coreference resolution solved by relaxation labeling. The approach combines the strengths of groupwise classifiers and chain formation methods in one global method. This system includes three phases:

Phase 1 - Graph representation

Let $G = G(V, E)$ be an undirected graph. Each mention m_i in a document is presented as a vertex $v_i \in V$ in G. An edge $e_{ij} \in E$ is added to the graph for pairs of vertices (v_i, v_j) representing the possibility that both mentions corefer. A subset of constraints $C_{ij} \in C$ is used to compute the weight value w_{ij} of the edge connecting v_i and v_j.

Phase 2 - Training process

Each mention pair (m_i, m_j) in training document is evaluated by the set of feature functions which form a positive example if the mention pair corefers, and a negative otherwise. For each type of mention m_j (for example: pronoun, named entity or nominal), a decision tree is generated and a set of rules is extracted with C4.5 [92] rule-learning algorithm.

Given the training corpus, the weight of a constraint C_k is related with the number of examples where the constraint applies and how many of them corefer.

Phase 3 - Resolution

The resolution algorithm solves the weighted constraint satisfaction problem dealing with the edge weights w_{ij}. In this manner, each vertex is assigned to a partition satisfying as many constraints as possible. The algorithm assigns a probability for each possible label of each variable (corresponding to each vertex in G). The process updates the weights of the labels in each step until convergence. Finally, the assigned label for a variable is the one with the highest weight.

SUCRE system

SUCRE [52] developed a feature engineering which can help reducing the implementation effort for feature extraction. It has a novel approach to model an unstructured text corpus in a structured framework by using a relational database model and a regular feature definition language to define and extract the features.

In learning, there are four classifiers integrated in SUCRE: Decision tree, Naive bayes, Support vector machine and maximum entropy. However, finally the best reported results were achieved with Decision tree. In decoding,the coreference chains are created. The system uses best-first clustering. It searches for the best predicted antecedent from right-to-left starting from the end of the document.

TANL-1 system

TANL-1 [3] was built based on highest entity-mention similarity. The authors applied Maximum Entropy classifier to determine whether two mentions refer to the same entity. The classifier is trained using the features extracted for each pair of mentions. If the pairwise classifier assigns a probability greater than a given threshold to the fact that a new mention belongs to a previously identified entity, it is assigned to that entity. In the case that more than one entity has a probability greater than the threshold; the mention is assigned to the one with the highest probability by using best-first clustering strategy.

UBIU system

Classification in UBUI [128] was based on mention pairs. UBIU used a combination of machine learning, in the form of memory-based learning (MBL) in the implementation of TiMBL [23], and language independent features. MBL uses a similarity metric to find the k nearest neighbors in the training data in order to classify a new example.

Despite of the difference in feature engineering, learning methods and some processing techniques, it can be seen that three later systems - SUCRE, TANL-1, and UBIU - belong to the approach called pairwise approach. The typical machine learning approach of these three systems includes two steps:

- Classification: systems evaluate whether each pair of mentions is coreferent with each other.
- Formation of coreference chain: Given the previous classification, the systems form coreference chain (mostly based on best-first clustering).

The approach presented in UBIU system joined classification and chain formation into the same step. In this manner, decisions are taken considering the whole set of mentions, ensuring consistency and avoiding that classification decisions are independently taken.

3.4.3 Evaluation metrics

In all experiments, we evaluated our system using closed gold-standard setting. It means that we used the gold-standard columns with true mention boundaries and our system was built strictly with the information provided in the task datasets. This is because our

system focuses on evaluating various approaches of previous participating systems versus our proposed listwise approach.

To evaluate our system, we also relied on four metrics which are MUC [116], BCUB [8], CEAF [61] and BLANC scores [98] provided by this shared task. The first three measures have been widely used, while BLANC is a proposal of a new measure interesting to test.

MUC-6/7 ([116])

This is the oldest and most widely-used metric which is based on coreference links. First, we count the number of common links between the reference (or "truth") and the system output (or "response"). The link precision is the number of common links divided by the number of links in the system output, and the link recall is the number of common links divided by the number of links in the reference.

BCUB ([8])

The MUC metric yields unintuitive results because of two main shortcomings. First, it does not give any credit for single-mention entities since no link can be found in these entities. Second, all errors are considered to be equal because in some tasks, some coreference errors do more damage than others. These drawbacks lead to the proposal of BCUB metric. This metric first computes a precision and recall for each individual mention, and then takes the weighted sum of these individual precisions and recalls as the final metric. The choice of the weighting scheme is determined by the task for which the algorithm is going to be used.

CEAF ([61])

The BCUB metrics still has its own problems: for example, the mention precision/recall is computed by comparing entities containing the mention and therefore an entity can be used more than once. Thus, they proposed Constrained Entity-Aligned F-measure or CEAF metric. It finds the best one-to-one mapping entities between the subsets of reference and system entities. They are aligned by maximizing the total entity similarity under the constraint that a reference entity is aligned with at most one system entity, and vice verse. After that, it computes the recall, precision and F-measure.

BLANC ([98])

BLANC is a measure obtained by applying the Rand index (Rand 1971) to coreference resolution and taking into account the shortcomings of the above previous metrics. The Rand index seems to be especially adequate for evaluating coreference since it allows us to measure 'non-coreference' as well as coreference links. Despite its shortcomings, it addresses to some degree the drawbacks of the previous metrics.

3.4.4 Feature sets

In this task, the feature sets were selected from the feature pool presented in [95]. We selected 22 features which are divided into three groups as described in more detail in Table 3.3.

Table 3.3: The feature sets used for all four languages). Non-relational features take on a value of **Y**ES or **N**O. Relational features indicate whether they are **C**OMPATIBLE, **I**NCOMPATIBLE or **N**OT **A**PPLICABLE.

Features describing m_j, a candidate antecedent	
1. PRONOUN_1	Y if m_j is a pronoun; else N
2. SUBJECT_1	Y if m_j is a subject; else N
3. NESTED_1	Y if m_j is a nested NP; else N
Features describing m_k, the mention to be resolved	
4. NUMBER_2	SINGULAR or PLURAL, determined using a lexicon
5. GENDER_2	MALE, FEMALE or UNKNOWN, determined using a list of common first names
6. PRONOUN_2	Y if m_k is a pronoun; else N
7. NESTED_2	Y if m_k is a nested NP; else N
8. SEMCLASS_2	The semantic class of m_k
Features describing the relationship between m_j and m_k	
9. HEAD_MATCH	C if the mentions have the same head noun; else I
10. STR_MATCH	C if the mentions are the same string; else I
11. SUBSTR_MATCH	C if one mention is a substring of the other; else I
12. NUMBER	C if the mentions agree in number; I if disagree; NA if numbers for one or both mentions cannot be determined
13. GENDER	C if the mentions agree in gender; I if disagree; NA if genders for one or both mentions cannot be determined
14. AGREEMENT	C if the mentions agree in both gender and number; I if they disagree in both number and gender; else NA
15. BOTH_PRONOUNS	C if both mentions are pronouns; I if neither are pronouns; else NA
16. SEMCLASS	C if the mentions have the same semantic class; I if they don't; NA if the semantic class information for one or both mentions cannot be determined
17. DISTANCE	Binned values for sentence distance between the mentions
Additional features describing the relationship between m_j, m_k	
18. NUMBER'	The concatenation of the NUMBER_2 feature values of m_j and m_k
19. GENDER'	The concatenation of the GENDER_2 feature values of m_j and m_k
20. PRONOUNS'	The concatenation of the PRONOUN_2 feature values of m_j and m_k
21. NESTED'	The concatenation of the NESTED_2 feature values of m_j and m_k
22. SEMCLASS'	The concatenation of the SEMCLASS_2 feature values of m_j and m_k

The first feature group encodes an antecedent candidate m_j. These features represent whether m_j is a pronoun, a subject of a sentence, or a nested noun phrase or not. The second group encodes the mention to be resolved m_k. These include the gender, number and semantic class of m_k, whether m_k is a pronoun and a nest noun phrase or not. This group also encodes the relationship between m_j and m_k. For example, whether two mentions have the same head noun, the same string, the same gender, the same number and so on. The third group encodes the additional relationship between the pair of an antecedent candidate and the mention to be resolved. These are the concatenation of the number, the gender, the pronoun information, the nested noun phrase and the semantic class feature values of m_j and m_k. These features are popular and available in all languages of the SemEval-2010 shared task 1 except for the semantic class of German language.

3.4.5 Comparing the listwise approach with previous participating systems

In this sub-section, we compare the performance of the listwise approach to participating systems of the SemEval-2010 shared task 1 including (1) RelaxCor system [102]; (2) SUCRE system [52]; (3) TANL-1 system [3]; and (4) UBIU system [128]. These are four systems that have the same experimental setting as in this work. We conducted experiments on four languages: English, Catalan, Spanish, and German. On each language,

Table 3.4: Experimental results of the proposed models on English (P: precision; R: recall; F1: F-score)

Types	Systems	MUC			BCUB			CEAF			BLANC		
		R	P	F1	R	P	F1	R	P	F1	R	P	F1
Participating Systems	RelaxCor	21.9	72.4	33.7	74.8	97.0	84.5	75.6	75.6	75.6	57.0	83.4	61.3
	SUCRE	68.1	54.9	**60.8**	86.7	78.5	82.4	74.3	74.3	74.3	77.3	67.0	70.8
	TANL-1	23.7	24.4	24.0	74.6	72.1	73.4	75.0	61.4	67.6	51.8	68.8	52.1
	UBIU	17.2	25.5	20.5	67.8	83.5	74.8	63.4	68.2	65.7	52.6	60.8	54.0
RankingSVM	Joint_MD_CR	40.6	63.4	49.5	79.2	89.8	84.2	77.4	77.9	77.7	69.8	70.2	70.0
	Threshold θ	45.8	57.4	51.0	80.6	87.7	84.0	76.7	77.4	77.0	73.9	75.7	74.8
Proposed Systems	ListNet	48.6	62.4	54.7	81.3	89.2	**85.1**	78.2	79.0	**78.6**	73.8	77.9	**75.7**
	ListMLE	64.3	49.9	56.2	86.2	77.3	81.5	73.4	83.3	**78.0**	84.0	73.6	**77.8**

we measured the performance of all systems in four evaluation metrics. In all tables presenting the experimental results, we make the F-score bold if it got the highest result in the same evaluation metrics. For ease of observation, we also make the F-score bold if our proposed systems got the higher result than results of participated systems (or the baseline model) in the same evaluation metrics.

Tables 3.4-3.7 show experimental results of four languages using four evaluation metrics. For each language, we provide results in three system types which are participating systems, RankingSVM systems, and our proposed systems. This subsection uses the first and the third types to analyze the effectiveness of the listwise approach over participating systems. The next Sub-section uses the second types to show the effectiveness of joining discourse-new detection to coreference resolution using threshold θ. Sub-section 4.7 uses the second and the third types to compare the proposed systems with a pairwise learning-to-rank baseline (RankingSVM).

English

Experimental results on English language are presented in Table 3.4. When using ListNet method, our system got the best results on three F-scores of BCUB, CEAF, and BLANC, and got the second best on MUC F-score. When using ListMLE method, our system got the best results on two F-scores of CEAF and BLANC and got the second best on MUC F-score. In that, CEAF and BLANC F-scores increased significantly from the previous highest scores: 75.6 to 78.6 (of ListNet) and 78.0 (of ListMLE) in the case of CEAF, and from 70.8 to 75.7 (of ListNet) and 77.8 (of ListMLE) in the case of BLANC. Our system also outperformed two systems of TANL-1 and UBIU in all four metrics. In comparison to SUCRE system, we achieved the comparative performance.

Catalan

Experimental results on Catalan are presented in Table 3.5. When using ListMLE method, our system got the best results on BLANC and MUC F-scores. It beats SUCRE, TANL-1 and UBIU systems in all four F-scores. In comparison with RelaxCor system, MUC and BLANC F-score increased significantly from 42.5 to 57.4 and from 59.7 to 70.5; CEAF and BCUB F-score decreased from 70.5 to 70.4, and from 79.9 to 77.1. This decrease is not remarkable in comparison with the increase of the MUC and BLANC metrics.

Table 3.5: Experimental results of the proposed models on Catalan (P: precision; R: recall; F1: F-score)

Types	Systems	MUC			BCUB			CEAF			BLANC		
		R	P	F1	R	P	F1	R	P	F1	R	P	F1
Participating Systems	RelaxCor	29.3	77.3	42.5	68.6	95.8	**79.9**	70.5	70.5	**70.5**	56.0	81.8	59.7
	SUCRE	51.4	58.4	56.2	76.6	77.4	77.0	68.7	68.7	68.7	72.4	60.2	63.6
	TANL-1	17.2	57.7	26.5	64.4	93.3	76.2	66.0	63.9	64.9	52.8	79.8	54.4
	UBIU	8.8	17.1	11.7	47.8	76.3	58.8	46.6	59.6	52.3	51.6	57.9	52.2
RankingSVM	Joint_MD_CR	35.7	46.7	40.5	70.6	78.5	74.3	65.2	65.2	65.2	66.7	61.8	63.8
	Threshold θ	40.2	55.7	46.7	71.8	84.0	77.4	67.8	67.9	67.8	66.6	71.1	68.5
Proposed Systems	ListNet	55.3	55.6	55.4	77.1	75.6	76.4	67.1	67.2	67.2	70.8	64.7	**67.2**
	ListMLE	58.7	56.1	**57.4**	78.5	75.8	77.1	70.4	70.5	70.4	73.4	68.3	**70.5**

Table 3.6: Experimental results of the proposed models on Spanish (P: precision; R: recall; F1: F-score).

Types	Systems	MUC			BCUB			CEAF			BLANC		
		R	P	F1	R	P	F1	R	P	F1	R	P	F1
Participating Systems	RelaxCor	14.8	73.8	24.7	65.3	97.5	**78.2**	66.6	66.6	66.6	53.4	81.8	55.6
	SUCRE	52.7	58.3	55.3	75.8	79.0	77.4	69.8	69.8	**69.8**	67.3	62.5	64.5
	TANL-1	16.6	56.5	25.7	65.2	93.4	76.8	66.9	64.7	65.8	52.5	79.0	54.1
	UBIU	9.6	18.8	12.7	46.8	77.1	58.3	45.7	59.6	51.7	52.9	63.9	54.3
RankingSVM	Joint_MD_CR	43.8	50.2	46.8	73.2	75.3	74.4	66.3	66.3	66.3	68.6	58.3	61.2
	Threshold θ	40.8	50.6	45.1	72.1	79.1	75.5	66.3	66.3	66.3	67.3	63.2	65.0
Proposed Systems	ListNet	58.2	57.5	**57.8**	78.5	75.9	77.8	69.1	69.2	69.2	71.7	64.6	**67.4**
	ListMLE	55.7	56.5	**56.1**	77.9	77.6	77.8	69.5	69.5	69.5	69.5	66.9	**68.1**

Spanish

Experimental results on Spanish language are presented in Table 3.6. Our system got the best results on the two F-scores of BLANC and MUC. For the two remaining F-scores of CEAF and BCUB, the proposed system's results are comparative to previous best scores (77.8 of both ListNet and ListMLE in comparison with 78.2 in the case of BCUB; 69.2 (of ListNet) and 69.5 (of ListMLE) in comparison with 69.8 in the case of CEAF).

Our system outperformed TANL-1 and UBIU systems in all four metrics. Compared with RelaxCor, our system got higher results on three F-scores of CEAF (from 66.6 to 69.2 of ListNet and 69.5 of ListMLE), MUC (from 24.7 to 57.8 of ListNet and 56.1 of ListMLE) and BLANC (from 55.6 to 67.4 of ListNet and 68.1 of ListMLE) which are all significant increase. For the remaining BCUB F-score, our system decreased insignificantly (from 78.2 to 77.78). Compared to the SUCRE system, our system got the comparative performance.

German

Experimental results on German language are presented in Table 3.7. ListNet system yielded the best results for BLANC metric and comparative results to SUCRE and TANL-1 systems. In comparison to SUCRE system, ListNet system decreased ≈14% in MUC score but it increased three remaining metrics (more than 2% in BCUB, ≈4% in CEAF and ≈5% in BLANC). In comparison to TANL-1 system, ListNet system significantly increased MUC(≈18%) and BLANC (≈14%) metrics and insignificant decreased BCUB (2.4%) and CEAF (1.1%) metrics. Our systems outperformed UBIU system on all four

Table 3.7: Experimental results of the proposed models on German (P: precision; R: recall; F1: F-score).

Types	Systems	MUC			BCUB			CEAF			BLANC		
		R	P	F1	R	P	F1	R	P	F1	R	P	F1
Participating Systems	SUCRE	74.4	48.1	**58.4**	90.4	73.6	81.1	72.9	72.9	72.9	78.2	61.8	66.4
	TANL-1	16.4	60.6	25.9	77.2	96.7	**85.9**	77.7	77.7	**77.7**	54.4	75.1	57.4
	UBIU	22.1	21.7	21.9	73.7	77.9	75.7	67.4	68.9	68.2	60.0	77.2	64.5
RankingSVM	Joint_MD_CR	29.6	41.7	34.6	80.1	86.2	83.1	75.9	76	75.9	63.5	57.9	59.9
	Threshold θ	39.4	35.4	37.3	82	79.4	80.7	71.5	71.6	71.6	65.7	61.2	63.1
Proposed Systems	ListNet	43.8	43.5	43.7	83.2	83.8	83.5	76.6	76.7	76.6	71.1	70.8	**71**
	ListMLE	40.4	44.4	42.2	80	89.2	84.3	76.3	76.4	76.3	64	69.1	66.1

evaluation metrics. The result also showed that ListNet performed slightly better than ListMLE system. However, in comparison to the best SUCRE system, our listwise approach got slightly lower performance.

3.4.6 Comparing the effect of joining discourse-new detection to coreference resolution

This subsection presents experimental results for comparing two settings of the anaphoricity detection on a model using RankingSVM (as shortly described in the resolution phase of Subsection 3.3). The first setting is the joint model of [95] in which it provides each active mention with the option to start a new cluster (new entity in our method) by creating an additional instance that (1) contains features that solely describe the active mention, and (2) has the highest rank value among competing candidates (i.e., 1) if it is discourse-new and the lowest rank value (i.e., 0) otherwise. In testing, if the additional test instance is assigned the higheset rank value by the ranker, then the active mention is classified as discourse-new. Otherwise, it is linked to the candidate that has the highest rank. This model is marked as RankingSVM (Joint_MD_CR) in Tables 3.4-3.7.

Another setting which we used to conduct all experiments of this chapter is using a threshold θ to determine the anaphoricity. If all candidates are assigned scores below the threshold θ, it means that the active mention is discourse-new and will not be resolved. Otherwise, it is linked to the candidate that has the highest rank. This model is marked as RankingSVM (Threshold θ) in Tables 3.4-3.7.

Experimental results showed that Joint_MD_CR system yielded slightly higher BCUB and CEAF metrics (0.2% and 0.7% respectively) but lower MUC and BLANC metrics (1.5% and 4.8% respectively) on English. On German, it yielded higher results on BCUB and CEAF (2.4% and 4.3% respectively) but lower MUC and BLANC metrics (3.3% and 3.2% respectively). On Spanish, it yielded higher MUC metric (1.7%) but lower results for three remaining metrics. It yielded the lower result on all four evaluation metrics on Catalan language. Therefore, we can see that on the copora of the SEMEVAL-shared task 1, the method of using a threshold to determine the anaphoricity is more effective. From this conclusion, we would like to choose this method to conduct all experiments of the ranking approach on all four languages of the shared task.

Table 3.8: Model names and their properties.

Model Name	Model Type	Method
RelaxCor	classification	groupwise
SUCRE	classification	-
TANL-1	classification	-
UBIU	classification	-
RankingSVM	ranking	pairwise
ListNet	ranking	listwise
ListMLE	ranking	listwise

3.4.7 Comparing the proposed models with a pairwise learning-to-rank baseline model

In this sub-section, we compare our listwise approach to a baseline model which was not implemented by any previous participating systems. It is the work of Ng and Rahman [95] which is a quite strong baseline model. This baseline model also exploits the learning-to-rank approach but using a pairwise method instead of using a listwise one. It belongs to the mention-ranking models. The baseline model exploited SVM^{rank} [46][4], an instance of SVM^{struct} for efficiently training Ranking SVMs [45], as the learning method. In this experiment, we use the same feature sets as in conducting experiments for ListNet and ListMLE models to ensure a fair comparison. The feature set is extracted as described in Table 3.3. For this use of the SVM learner in our experiments, we set all parameters to their default values as the same as the experimental setup in the work of [95]. For choosing the best parameter θ, we also tune the parameter on the development set.

Table 3.8 shows properties of models used in our experiments. We can see that all participating models in SemEval-2010 shared task 1 belong to the type of classification, the RankingSVM belongs to the pairwise ranking approach and our two models proposed in this chapter belong to the listwise ranking approach. Among three ranking models, ListNet and ListMLE are listwise models, while RankingSVM is a pairwise model.

Tables 3.4-3.7 also contain the experimental results using three learning-to-rank methods. Experimental results showed that the listwise approach outperformed the baseline pairwise approach. For all four languages, each of our proposed systems got the higher results on most of four evaluation metrics in comparison with the pairwise approach. Considering the remaining metrics which our systems could not outperform, it is the fact that the decrease is not remarkable in comparison with the increase of other scores. This leads to the sum of four metrics for each languages increases when using the listwise approach. In other words, the listwise approach gave the better result compared to the ranking model that uses a pairwise approach.

3.4.8 Comparing two methods of getting training instances

To verify that our learned model is not biased to anaphoric mentions with a large number of candidates as in previous approaches, we also conducted experiments to compare two ways of generating training instances.

[4]This software can be downloaded from
http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

1. Subset_of_Preceding_Mentions: We only consider a subset of preceding mentions (candidates) which include one closest positive example and all negative examples between the mention and the closest positive one.

2. All_Preceding_Mentions: We consider all preceding mentions.

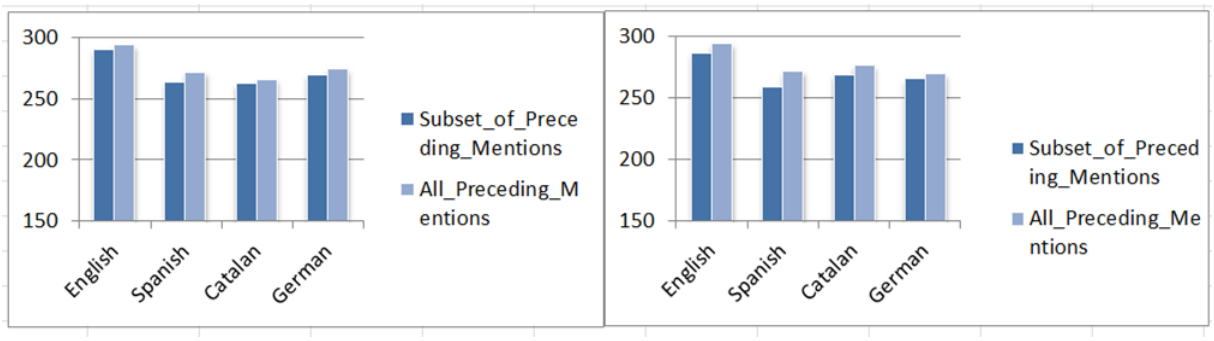

Figure 3.3: Sum of four metrics on listwise learning-to-rank methods (On the left: ListNet; On the right: ListMLE).

Figure 3.3 shows the chart of sums of four metrics for all four languages using the listwise learning-to-rank approach. The experimental results on both two listwise algorithms ListNet and ListMLE showed that using all candidates to generate training instances yields the better results in all four metrics in comparison to using only a subset of candidates. This means that if we provide more context for training, it helps the ranker learn better parameters for the coreference resolution task.

3.4.9 Some more results

P-R curves

This part presents P-R curves (see Figure 3.4) that give a more informative picture of the system's performance. We chose ListNet system as an example to conduct survey. Each P-R point corresponds to a different value of the threshold θ. For the 3 metrics: MUC, BCUB, and BLANC, the threshold is varied incrementally by the value of 0.01, with consideration in the best threshold (the threshold which results in the highest sum of four metrics) is in the middle points of curves. From the lower threshold to the upper threshold, the P and R of the CEAF metric got the max value at the middle values.

The graph indicates that on all four languages, MUC usually got the lowest results and BCUB usually got the highest results. CEAF and BLANC got the intermediate levels. In all four languages, precision increases when recall increases on CEAF metric. In three remaining metrics, it can be seen that precision usually decreases when recall increases or vice verse. Therefore, we need a tradeoff between them depending on real applications.

Some true examples

This sub-section shows some English examples that listwise approach using ListMLE can correctly determine the antecedent while the pairwise approach using SVM^{rank} cannot. In the figures 3.5 and 3.6, we used color texts and arrows to make them easy to understand. The blue texts show referring expressions of the entity which our proposed method

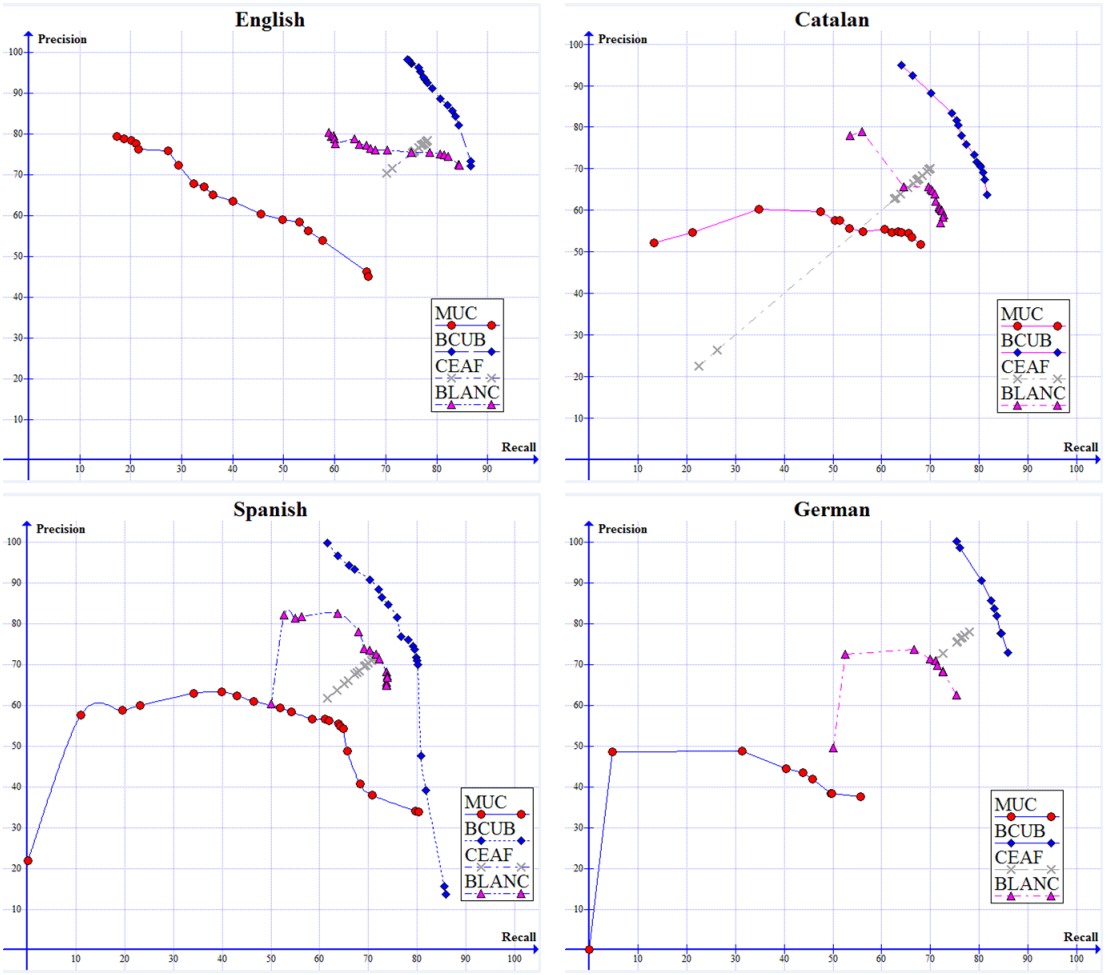

Figure 3.4: P-R curves on four evaluation metrics of four languages.

correctly determine and the red texts show the one that the baseline model wrongly guest for a mention. The blue arrows express the correct mention-antecedent pairs while the red arrows express the wrong pairs. A mention pointing to the other is an anaphor while a mention pointed by the other is an antecedent.

3.4.10 Discussion

The nature of previous approaches limits the number of antecedent candidates that can be considered in a candidate set. If we enlarge the number of candidates, the number of generated training instances varies largely between anaphoric mentions, which will result in training a model biased toward anaphoric mentions with more candidates. Our listwise approach does not suffer from this limitation. Moreover, experimental results in Subsection 4.7 showed that a better performance was achieved when we increase the size of the candidate set used in the training phase. Another advantage is that our approach considers all antecedent candidates simultaneously, so it does not hold the strong assumption about the i.i.d of candidates or candidate pairs. It provides us a more natural approach that is suitable to the fact that in a given context of a document, mentions are related to each other in some views; therefore, it is better to consider them simultaneously. In addition, with the usage of listwise loss functions, the performance showed that the listwise approach is mostly more effective than previous approaches for the task

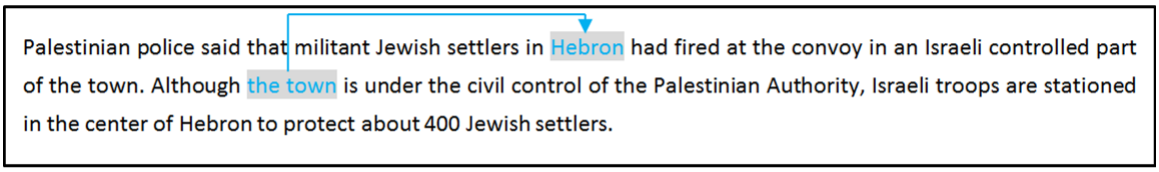

Figure 3.5: This example shows that our method correctly determined the antecedent for the mention *the town*. While the baseline pairwise method cannot find this antecedent and therefore determined this mention is non-anaphoric.

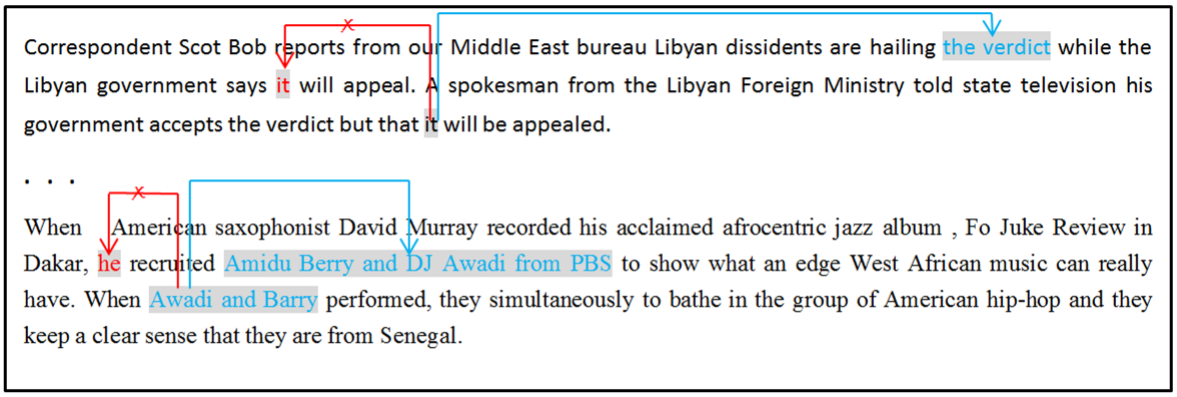

Figure 3.6: This case shows two examples in which the listwise method correctly determines the antecedent for each mention while the baseline pairwise method could not.

of coreference resolution in the corpora of the SemEval-2010 shared task number 1. We achieved the comparative performance in comparison with the best participating system SUCRE, which use the Decision Tree algorithm with best-first strategy.

In comparison with the pairwise learning-to-rank approach, experimental results showed that the listwise method yields better results than the pairwise method. Among listwise learning-to-rank approaches, the system using ListMLE performed slightly better than the system using ListNet on English and Catalan, but slightly poorer on Spanish and German. Experimental results also showed that using all negative and positive examples to create training instances gives better results than using only negative examples between the closest positive example and a given mention as training instances. In addition, our model also act as an additional filter for detecting anaphora using the threshold θ.

Among lots of metrics proposed for evaluating a coreference resolution system, none of them is fully adequate. Each metrics has its own strong points as well as weak points as we discussed in section 4.1. This situation makes it hard to successfully compare different systems. Getting the state-of-the-art performance on these four common metrics seems to be a very difficult task. Until now there is no common agreement on a standard measure for coreference resolution task. However, based on formulas and characteristics of each metric, it is common that later-proposed metrics usually give the better quality than metrics proposed early. If using this criterion, we saw that we obtained the highest F-score for the latest proposed BLANC metric in all four languages.

3.5 Conclusion

In this chapter, we presented an empirical study on using a listwise approach to coreference resolution. We formulated the task as a learning-to-rank problem and then exploited a listwise learning-to-rank approach to solve the task. In this listwise approach, instead of using single candidates or pairs of candidates, we use lists of candidates as training instances. This approach allows all candidate antecedents to be considered simultaneously in both the training and the resolution phases, which yields more benefits than traditional approaches. All experiments presented in this chapter were conducted and compared in the closed gold-standard setting of public corpora of the SemEval-2010 shared task 1, the task of *Coreference Resolution in Multiple Languages*. The experimental results showed that the proposed approach obtained relatively good performance in all four languages including English, Catalan, Spanish, and German. For the latest proposed evaluation metric BLANC, our models got the highest results in the F1 score. In comparison with a pairwise learning-to-rank approach, our approach yielded a better performance. These results suggest that this listwise approach is promising for the coreference resolution task in multiple lanuages.

Chapter 4

Automated Reference Resolution in Legal Texts

Reference resolution in legal texts is a new interesting task in the Legal Engineering research. The goal of this chapter is to create a system which can automatically detect references and then extracts their referents. Previous work limits itself to detect and resolve references at the document targets. In this chapter, we go a step further in trying to resolve references to sub-document targets. Referents extracted are the smallest fragments of texts in documents, rather than the entire documents that contain the referenced texts.

4.1 Introduction

Legal Engineering [49, 50, 51] is a new research field which aims to achieve a trustworthy electronic legal society. The main goals are to help experts make complete and consistent laws and to design an information system which works based on laws. Hence, it is vital to develop a system which can process legal texts automatically. One of the obstacles to law processing is that, at the discourse level, legal texts contain many reference phenomena. These references usually contain precious information. The law will be difficult to comprehend if we cannot access the referenced items within it. Resolving the reference phenomena, therefore, is an important task in Legal Engineering.

Figure 4.1 shows excerpts from three documents[1] named A12P1, A12P4, and A13P1. These excerpts contains two references[2] (the texts bounded in red angle brackets), i.e. '*the provision of previous article, para 4*' in the document A13P1 and '*the notification in the provision of para 1*' in the document A12P4. To comprehend the contents of these documents, it is important to know the referenced items. In other words, we need to know to which part of texts (the texts bounded in green square brackets) these references refer. This kind of references is very popular in legal documents because law-makers usually import pieces of available information which have already been introduced in other documents by using briefer expressions. This, as a result, helps to guarantee the soundness as well as the consistency in a law system. We name these briefer expressions

[1]The term '*documents*' corresponds to articles, paragraphs, items, or sub-items according to the naming rules used in the legal domain.

[2]These two reference examples are two typical examples of two classes of references, which will be described in more detail later.

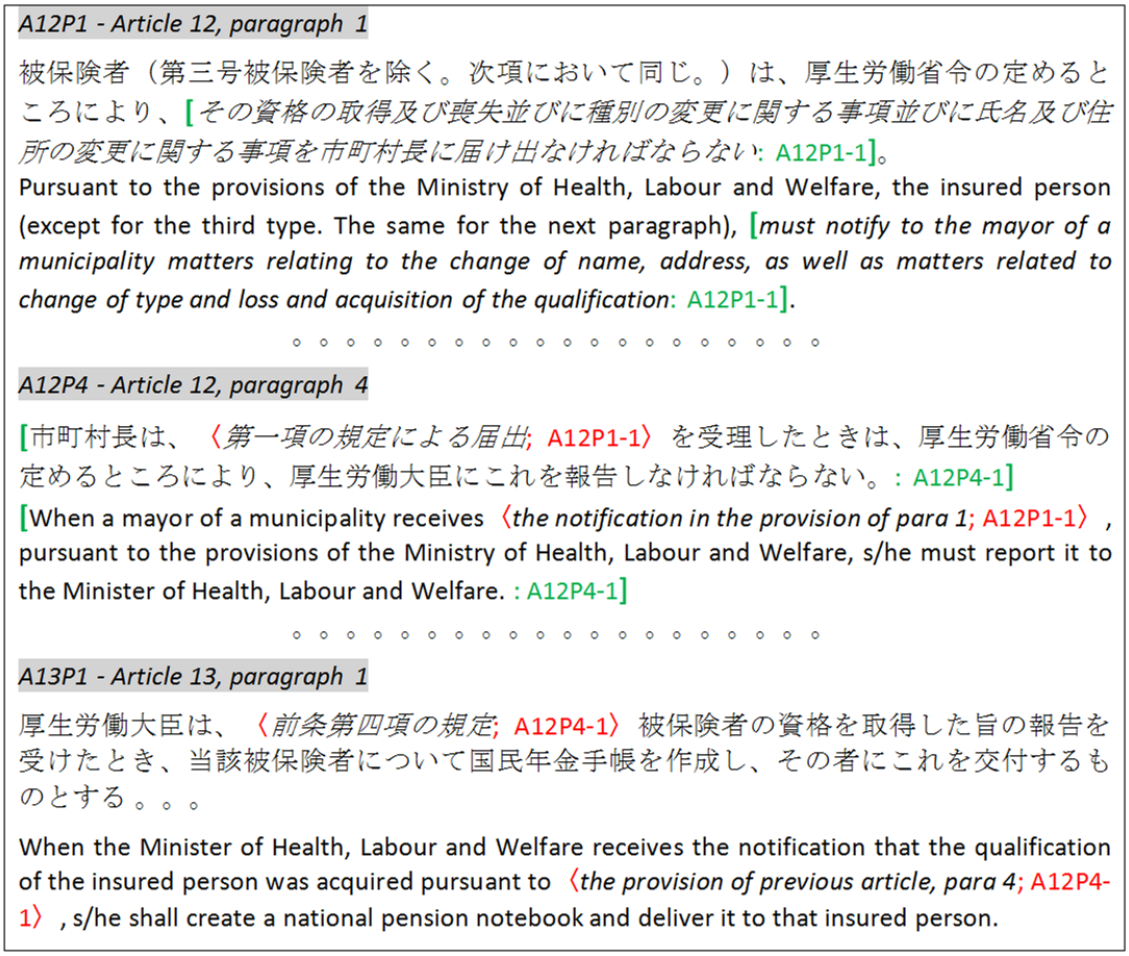

Figure 4.1: Examples of reference phenomena in legal texts (In this figure, references are bounded in red angle brackets (⟨⟩) while their referents are bounded in green square brackets ([ ]).). Expressions start after a colon (:) or a semicolon (;) in the bounded texts are the identification expressions (ID) of these texts (i.e. A12P1-1). A reference and its referent have the same ID

.

'*references (or mentions)*', and their referenced items '*referents (or antecedents)*'. We will discuss the characteristics of references as well as their referents later in Section 4.3.

There exists some work related to this kind of research, developed for several languages such as Italian [12, 82], Spanish [68], and Dutch [64]. In previous work, authors focus on detecting and resolving so-called normative references to distinguish them from the above references. Normative references are slightly different from the above references. In the above examples, normative references would appear in the forms of '*para 1*' and '*previous article, para 4*'. In resolving these normative references, authors limit resolvers to identifying only the referred documents but not to which parts of texts in these documents[3]. Normative references are much easier to detect and resolve because most references are correct and unambiguous. They mostly follow some regular expressions. Therefore, knowing the naming rules of the law, we can easily determine the identification

[3]With this output, users/law-makers need to read over the referenced document to find which part of texts is actually referred to

of the document which is referred to by a reference.

In this chapter, we study the reference resolution task for a non-Western language, particularly the Japanese language. Different from previous work, which is limited to determine the positions of the referred documents (A12P1 and A12P4), in this research, we go a step further. Our reference resolver tries to extract the smallest fragments of texts that are actually referred to by references (the texts in green square brackets). Resolving this type of references is more difficult because it requires syntactic and semantic understandings of references and its context information as well as of the referenced document that contains the referenced texts. Therefore, the methods used in previous work are not sufficient to resolve these references. Based on the characteristics of reference phenomena in legal texts which will be discussed in Section 4.3, we propose a four-step framework using machine learning approaches which results in the final system being automatically trainable from a corpus with a minimal amount of human intervention. We recognize references (mentions) in the first step - Mention Detection. For each output reference, we then extract its contextual information in the second step - Contextual Information Extraction. This contextual information will be used in the third step - Antecedent Candidate Generation - to generate referent (antecedent) candidates of this mention. The fourth step - Antecedent Determination - will select the best one among its candidates to determine its exact referents.

Our main contributions can be summarized as follows:

- Introducing the task of reference resolution in the domain of legal texts, in which we detect references and then map them to the smallest fragment of texts that they refer to.
- Analyzing the characteristics of references in the legal texts. Based on this analysis, we propose a four-step framework using machine learning approaches to solve the task.
- Introducing a new annotated corpus, the Japanese National Pension Law (JNPL) corpus on reference resolution.
- Conducting experiments and evaluating our framework on the JNPL corpus.

From the experiments, we obtained 80.06% in the F1 score in detecting references and 85.61% accuracy in resolving them. In the whole system, we obtained 67.02% in the F1 score. Building a good reference resolver brings many potential benefits such as providing computer science supports in making, maintaining and validating legal documents; supporting in finding contradictions in legal texts, which is obtained by reasoning and semantic processing such as semantic parsing and reference resolution; supporting in building a question answering system that allows citizens to have easier access to legal information, etc.

The rest of this chapter is organized as follows: Section 4.2 reviews related work. Section 4.3 introduces some characteristics of reference phenomena in legal texts. Based on those characteristics, Section 4.4 presents our proposed framework to solve this task. Section 4.5 presents solutions for each step in the framework. Next, we will describe experiments in Section 4.6. In this section, we also analyze the impact of each step on the whole system and illustrate an output example of the final system. In addition, we propose a semi-supervised technique to improve the performance of the system. Section

4.7 provides an analysis of errors in the proposed framework. In Section 4.8, we discuss the performance and the versioning problem of the system. Finally, Section 4.9 concludes the chapter and proposes future work.

4.2 Related work

This section presents three kinds of related works, and places our work in the scope of research on references. Firstly, we review some typical studies which also consider resolving a fragment of texts to documents or sub-document targets. Secondly, we review some work on reference resolution conducted in general texts. Finally, we focus on studies on detecting and resolving references within the legal domain, which are closest to our work. In this final sub-section, we also distinguish how our work is different from the previous work.

4.2.1 Studies on resolving a fragment of texts to documents or sub-document targets

If we consider each reference as an anchor and its referent as a linked document, our work somehow can be regarded as citation linking which is studied in works such as Entity-linking, Linking-the-Wiki, citation processing, etc.

- *Entity-Linking* is a track on TAC KBP [4] which is a series of evaluations and workshops organized to promote research in Natural Language Processing and related applications. Given a name (of a Person, Organization, or Geopolitical Entity) and a document containing that name, the goal of this track is to determine the Knowledge Base node (KB) for the named entity, adding a new node for the entity if it is not already in the KB. The reference KB is derived from English Wikipedia, while source documents come from a variety of languages, including English, Chinese, and Spanish. Such systems typically search for candidate entities and then disambiguate them, returning either the best candidate or NIL. Some representative works are by [36] and [35].

- *Link-the-Wiki*[5] aims to produce a standard procedure and metrics for the evaluation of link discovery between documents. Given a new orphan (unlinked) document, the task is to analyze the text and recommend a set of outgoing links from anchors (specified as passages in the orphan document) to Best Entry Points (BEPs) in existing documents in the collection. The BEP for a link should be the position in the target document from which the reader, having just followed the link, should begin reading.

- *Citation Processing*: A common prerequisite for knowledge discovery is to accurately combine data from multiple, heterogeneous sources into a unified, mineable database. An important step in creating such a database is record deduplication, consolidating multiple records that refer to the same abstract entity (i.e. [22]).

[4] http://www.nist.gov/tac/2013/KBP/EntityLinking/index.html
[5] http://www.inex.otago.ac.nz/tracks/wiki-link/wiki-link.asp

4.2.2 Studies on reference and anaphora resolution in general texts

Reference resolution ([47]) is the task of determining which entities are referred to by which linguistic expressions. This task plays an important role in a large number of natural language processing (NLP) applications. Therefore, it has attracted much attention within the NLP community. Among all types of reference phenomena, coreference is the most popular and is the focus of most research on reference resolution in general texts. When the reader must look back to the previous context, reference is called 'anaphoric reference'. When the reader must look forward, it is termed 'cataphoric reference'. Many works on various aspects (linguistic features [37, 75],machine learning models [104], multiple languages [99], etc.) of the reference resolution in the general domain have been published. This has boosted the development of robust machine learning models for the task in the general domain.

Another type is discourse deixis [27, 30, 73], which relates an anaphor to a verbal or (multi-)clausal antecedent. Discourse entities are called abstract objects because they refer to propositional entities, such as events or facts. Grammatical rules used in CoRe cannot be applied to this case because the antecedent is non-nominal. Although abstract anaphora are not able to be reliably resolved in the near future, their identification would be an important prepocessing step in a general CoRe system. The performance of CoRe system will be improved if abstract anaphors are not assigned incorrectly to an NP antecedent.

4.2.3 Studies on reference resolution within the legal domain

This section describes studies on reference resolution of legal texts, which are the most closely related to our work. To the best of our knowledge, there are only a few studies done for Italian, Dutch, and Spanish.

For Italian, there exists two typical studies. The first one is the work of [82], in which the authors conduct a project to work out and implement a model for recognizing, understanding, normalizing the normative references found in legal texts and bringing such references under a set of common standards in order to favour the interoperability between different legal information systems. The second one is the work of [12], in which the authors name references as citations. On the basis of the different writing of the element Act, authors have classified the various citation formats provided for by the drafting rules into three categories, called: normal, simplified, and non-paradigmatic citations. They also model each type of citations to extract templates to detect them. The first analysis carried out on the laws enacted in the 1990s making up part of the selected legislative corpus confirms that, up until that time, it is possible to identify and extract more than 95% of the explicit textual legislative citations, conforming to the legislative drafting rules.

For Dutch, [64] also perform automated detection of reference structures in law. They first discuss the type and structure of references in Dutch legal sources, and then propose a grammar containing most of the patterns analyzed to recognize these references. After a reference has been found in a text, they resolve it to the URI of the document that is referred to. They also expand their work to other legal sources. In testing, they achieved a very high accuracy, between 95% and 99%.

For Spanish, [68] present an application of information extraction. Its goal is to automate the extraction of references from legal documents and the storage of their information in order to facilitate an automatic processing of these information items by services offered in digital libraries. They first classify references using four criteria. After that, they extract patterns based on the information about vocabulary used to name legal items and the grammar associated with each type of reference. They use these patterns to extract references for each input document. Each extracted reference is then processed by the solver, that tries to match each reference to some legal items.

In previous related works, the authors only consider references in terms of linguistic expressions that identify a specific act or a text partition referred to. Their purpose is to link the content of laws based on the naming rules used in a specific legal domain. With an input sentence of legal texts, which contains the phrase '*the notification in the provision of paragraph 1*', these works only consider extracting the reference '*the provision of paragraph 1*', and then resolve them to the entire referenced document (i.e. Paragraph 1 of Article 12). Hence, to understand which notification is referred to, users need to read that document to get its description. This is somewhat redundant because that document may contain unnecessary information for the comprehension of the input sentence.

In this research, we go a step further. To avoid over-reading these unnecessary texts in the referenced document, we consider a wider range of references. We extract references in the form of which content is referred to and the location containing that content. After that, we resolve them to the exact explanation of that content rather than only the entire document which covers the explanation. In particular, in the previous input sentence, we extract the full phrase and resolve it to the smallest fragment of texts that describes the type of the notification in Paragraph 1, i.e. '*notification of matters relating to the change of name, address, as well as matters relating to change of type and loss and acquisition of the qualification*'. Moreover, we propose the use of machine learning approaches (rather than rule-based approaches as used in the previous works) which results in the final system being automatically trainable from a corpus with a minimal amount of human intervention.

4.3 Characteristics of references in legal texts

Some characteristics of references in the legal domain are worth consideration when dealing with an automatic extraction and resolution of references. Most types of references in legal texts relate to terms, definitions or provisions of articles. Although referenced texts can occur before or after a reference, in this research we derive the term '*mention*' and '*antecedent*' from reference resolution in general texts to describe this relationship. We use the term '*mentions*' to denote linguistic expressions that contain referring texts. The texts that mentions refer to are called '*antecedents*'.

Mentions in legal texts have their own structures, which are different from mentions in the general domain. They mostly conform to several kinds of patterns. Figure 4.2 [6] shows that a mention usually consists of two main parts: a *position* part and a *content* part. The later part may be a noun or a noun phrase (This noun phrase can be nested). The former part usually conforms to a regular expression. By observing the position parts, we

[6] In Figure 4.2, | means 'or', [ ] means 'optional', and + means 'repeat one or more times'. An example of a mention and its translation into English are also given in Figure 4.2.

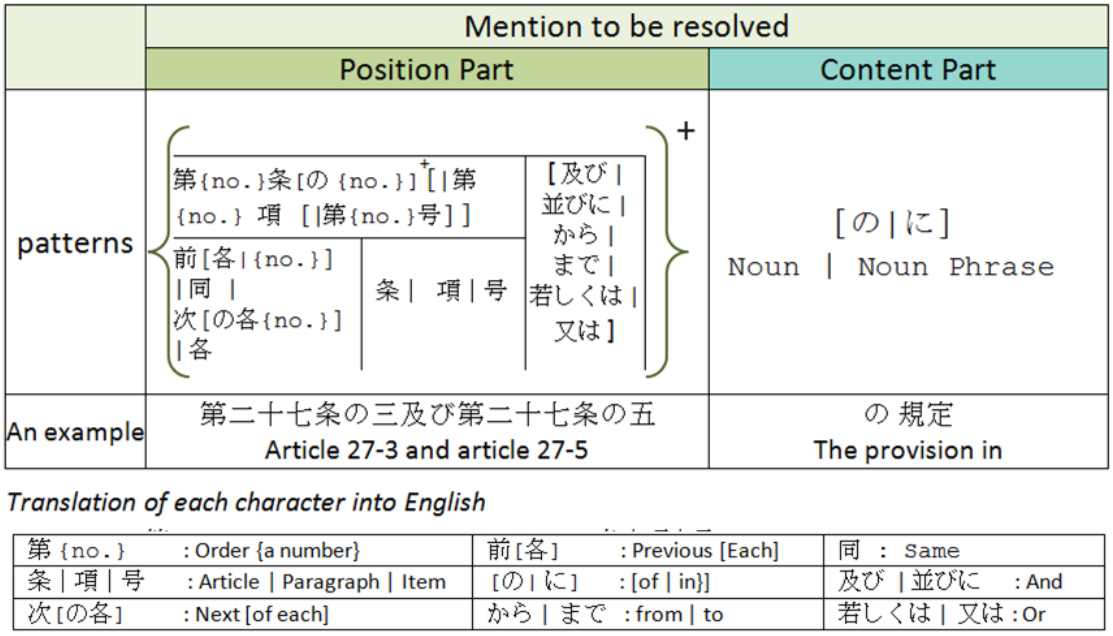

Figure 4.2: The structure of mentions in legal texts.

can see that they mostly fall under one of the following types:

1. Single position: concerns a single position of a mention.

   (a) Well-formed positions: These positions comprise a label, such as an article, a paragraph, or an item combined with a number. They usually start with the broadest scope and end with the narrowest one. These positions contain all the information needed to identify the location of the referred item.

   (b) Anaphora, indirect positions: They often refer to a position of an earlier mention. These positions are always resolved to one of the former (label and number). This requires some context information (documents referenced before and near the current mention, in a specific part of the same texts, ..) to solve the mention.

2. Coordinated positions: Several antecedents are referred to in the same mention by using some linking expressions such as *and, or, from ... to*, etc.

3. Special cases: These are used when an element in the text contains a list that is preceded by a description of the list, without which the list does not make sense. Another special case is the use of the word *each time* that refers to sub-items of an article.

Figure 4.3 shows some examples of position parts in mentions of legal texts, which belong to different types as distinguished above.

Antecedents usually are definitions or explanations of related terms or provisions. They can be nouns, noun phrases, sentences, and paragraphs of articles or they can even be whole articles in some cases. They help readers to comprehend the law thoroughly, and also help lawmakers to create legal texts that are concise and easy to understand.

No.	Position Parts	Types
1	第七条第一項第二号 -- Article 7, paragraph 1, Item 1	1) a)
2	同項 -- The same paragraph 前項 -- The previous paragraph	1) b)
3	次号及び第三号 -- The next and the third items	2)
4	第三条（第二項、第三項を除く。） -- Article 3 (excluding Paragraph 2, and Paragraph 3)	3)
5	次の各号のいずれか -- Each of the following items	3)

Figure 4.3: Some examples of different types of position parts of mentions in legal texts.

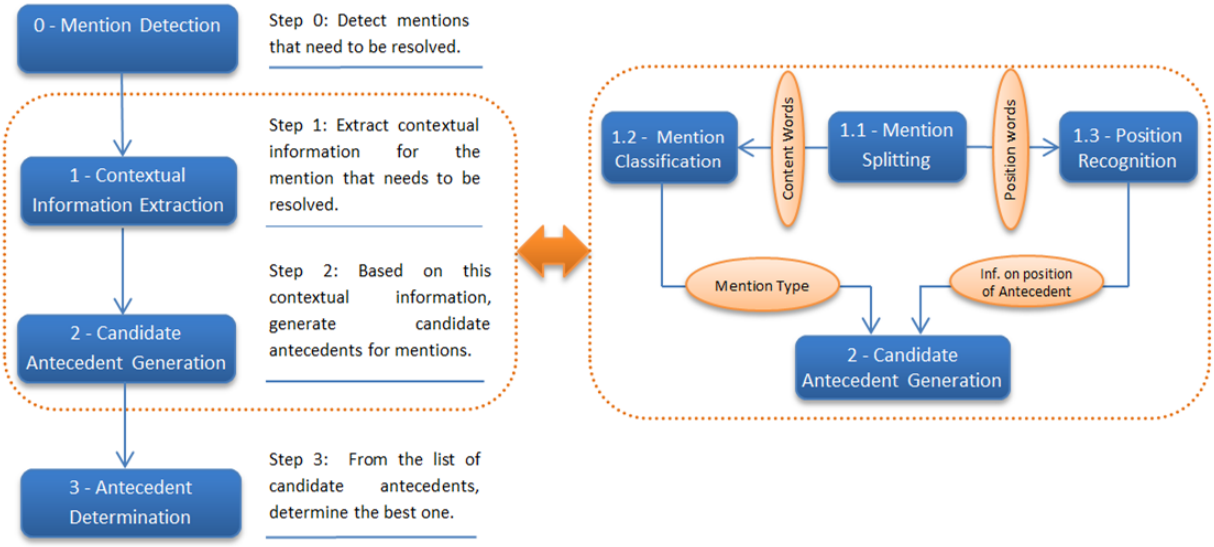

Figure 4.4: A four-step framework for resolving references in legal texts.

4.4 A four-step framework to reference resolution in legal texts

This section presents our proposed framework for the reference resolver as shown in Figure 4.4. In this figure, oval nodes denote data and box nodes denote processes. In comparison with reference resolution in general texts, this framework has two additional steps (Step 1 and Step 2 that are bounded in the dotted rounded-corner rectangle). These two steps take advantage of the characteristics of references in legal texts for solving the task more effectively.

Step 0 - Mention detection

This step identifies the occurrence of mentions in legal texts. Detecting mentions is very important for the downstream processing of the framework.

Each mention extracted in this step will be analyzed via two intermediate steps, viz Step 1 and Step 2. The purpose of Step 1 is to extract the context information of the mention which benefits the generation of its antecedent candidates in Step 2.

Step 1 - Context information extraction

The goal is to determine the context information, including the locative information that a given mention refers to, documents referenced before and near the current mention, the content part of the mention, and its classification. This information will be used later to narrow down the search space of antecedent candidates for that mention. This step is divided into three sub-steps as shown on the right hand side of Figure 5.3.

- 1.1 - Mention Splitting: Splits a mention into two parts which are a position part and a content part. These two parts are independent, and therefore should be processed separately. The position part allows to locate the position of the document containing the referenced texts while the content part helps in extracting the smallest fragment of texts that this mention refers to.

- 1.2 - Mention Classification: Determines whether a mention refers to an entire document (*Class 1*) or only a fragment of a document (*Class 2*). Depending on the classification of a mention, different strategies are designed to determine its antecedent.

- 1.3 - Position Recognition: Locates the scope of antecedents. The goal is to exactly determine which articles, which paragraphs, which items etc., a given mention refers to.

Step 2 - Antecedent candidate generation

This step uses the position information extracted from Step 1 to determine the scope of the exact document (in which articles, which paragraphs, which items) that a mention refers to. From that document, we generate antecedent candidates for a given mention based on the dependency trees of the antecedent sentences or punctuation marks such as commas and full stops.

Step 3 - Antecedent determination

From a list of candidates generated in Step 2, we have to determine an exact antecedent for each mention. The exact antecedent is the smallest fragment of texts that this mention refers to. This is the most important and the most difficult step. The strategy to determine the exact antecedent of a mention depends on the classification of a mention in the mention classification step (Sub-step 1.2). We will describe these strategies in more detail in Section 4.5.

4.5 Solutions to each step of the framework

In this section, we will present our proposed solutions for each step of the framework in the previous section.

4.5.1 Mention detection and mention splitting

In the mention detection step, the goal is to extract all mentions appearing in the input string. The input string is a sequence of words and the output is a collection of mentions (or references) contained in the input string. Each extracted mention will be an input for the mention splitting sub-step. This sub-step divides this mention into two parts: the position and the content parts.

Source Sentence	市町村長は、第一項の規定による届出を受理したときは、 When a mayor of a municipality receives *the notification in the provision of Para 1*,															
Word sequence	市町村	長	は	、	第	一	項	の	規定	による	届出	を	した	とき	は	、
	a mayor of a municipality				*the notification in the provision of Para 1*							When receives				
IOB notation	O	O	O	O	B_M	I_M	I_M	I_M	I_M	I_M	I_M	O	O	O	O	
IOE notation	O	O	O	O	I_M	I_M	I_M	I_M	I_M	I_M	E_M	O	O	O	O	
FIL notation	O	O	O	O	F_M	I_M	I_M	I_M	I_M	I_M	L_M	O	O	O	O	

Figure 4.5: Mention Detection: A law sentence in the IOB, IOE and FIL notations.

Similarly to many classical NLP tasks such as text chunking [90] and named entity recognition [32], we also formulate these two tasks as sequence labeling problems. Sequence labeling is a type of pattern recognition task that involves the algorithmic assignment of a categorical label to each member of a sequence of observed values. A common example of a sequence labeling task is part of speech tagging, which seeks to assign a part of speech to each word in an input sentence or document. Sequence labeling can be treated as a set of independent classification tasks, one per member of the sequence. However, accuracy is generally improved by making the optimal label for a given element dependent on the choices of nearby elements, using special algorithms to choose the globally best set of labels for the entire sequence at once.

In the mention detection step, each word is assigned a label indicating whether it starts a specific mention, is inside a specific mention, or is outside any mention. Figure 4.5 illustrates an example of a Japanese law sentence in the IOB, IOE and FIL notations. In this example, the source sentence contains one mention (red texts) that should be detected. The labels of this sentence using these three notations are described as follows:

- In the IOB notation, the first element of a mention is tagged with B_M (**B**egin of **M**ention); the remaining elements of the mention are tagged with I_M (**I**nner of **M**ention); all elements outside the mention are tagged with O (**O**thers).

- In the IOE notation, the first and the intermediate elements of a mention are tagged with I_M (**I**nner of **M**ention); the last element of the mention is tagged with E_M (**E**nding of **M**ention); all elements outside the mention are tagged with O (**O**thers).

- In the FIL notation, the first element of a mention is tagged with F_M (**F**irst of **M**ention); the intermediate elements of the mention are tagged with I_M (**I**nner of **M**ention); the last element of the mention is tagged with E_M (**E**nding of **M**ention); all elements outside the mention are tagged with **O** (**O**thers).

Each detected mention is split into two parts; the position part and the content part. Figure 3.5 illustrates an example of a mention in the IOB notation: the first word of the position part is tagged with B_P (**B**eginning of the **P**osition part); the remaining words of the position part are tagged with I_P (**I**nner of the **P**osition part); all words of the content part are tagged with O (**O**thers).

In these two tasks, we use Conditional Random Fields (CRF) [54] as the learning method to learn the sequence labeling models. CRFs are discriminative undirected graphical models, which encode known relationships between observations and construct consistent interpretations. It is often used for labeling or parsing of sequential data. There

Source Sentence	第一項の規定による届出 *the notification in the provision of Para 1*						
Word sequence	第	一	項	の	規定	による	届出
Parts	*Para 1*			*the notification in the provision of*			
	Position Part			Content Part			
Tag sequences	B_P	I_P	I_P	O	O	O	O

Figure 4.6: Mention Splitting: A law mention in the IOB notation.

No.	Mentions of class 1	Mentions of class 2
1	第九十四条第四項の規定 The provisions of Article 94 paragraph 4	前項の財政均衡期間 Balanced budget period of previous paragraph
2	第三号から第五号までのいずれか One of items from the item 3 to item 5	第一項の規定により保険料を納付する者 People who pay insurance premiums in accordance with the provisions of paragraph 1

Figure 4.7: Some examples of mentions of two classes.

are three popular reasons for using CRFs; (1) the nature of these two tasks; (2) the advantages of CRFs, which have all the strong points of Maximum Entropy Markov models [67] but do not suffer from the label bias problem; (3) CRFs have been applied successfully to many NLP tasks such as POS tagging, chunking, named entity recognition, syntax parsing, etc.

In extracting feature sets, we use a combination of n-gram (n $\leq$ 3) of words, part-of-speech tags, and chunking information. When performing experiments, we exploit three kinds of label settings which are the IOB, IOE, and FIL notations [60].

4.5.2 Mention classification

In this sub-step, each extracted mention of the previous step is classified into one of two predefined classes; *Class 1* and *Class 2*. Figure 4.7 shows some examples of mentions belonging to these two classes.

To achieve the goal of this sub-step, we train a classifier using supervised machine learning methods that have the advantage of combining arbitrary types of information in making a classification decision. We use two robust classifiers which have demonstrated strong performance in many classification tasks especially in statistical NLP. In the first classifier, the classification is performed with a statistical approach, built around the maximum entropy (MaxEnt) principle [97]. This principle allows to estimate the conditional probability $p(y|x)$ that a model outputs a label y given a context x as follows:

$$P(y|x) = \frac{1}{Z(x)} exp(\sum_i \lambda_i f_i(x, y))$$

where $f_i(x, y)$ refers to a feature function; λ_i is a parameter of the model; and $Z(x)$ is a normalization factor. There are several models that use this method, subject to certain constraints. Among these models, the MaxEnt chooses the model with the flattest probability distribution (corresponding to the highest entropy). This constrained optimization

problem is first converted into a dual optimization problem using Lagrange multipliers. The solution is then found by applying the improved iterative scaling method or LBFGS method.

In the second classifier, we use support vector machines (SVMs) - a statistical machine learning technique proposed by [115]. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. Besides linear classification, SVMs can also deal with non-linear classification efficiently by using kernel functions which implicitly map its inputs into high-dimensional feature spaces.

With each mention, we will extract features based on n-grams ($n \leq 3$) of words appearing in the mention. After that, the model will classify which class it belongs to. We set up two experimental settings by using two kinds of words: content words which are the output of the mention splitting step and all words of a mention (not only content words but also position words).

4.5.3 Position recognition

In this sub-step, the position of the antecedent of a given mention is located. The input is a position part, which is the output of the mention splitting sub-step. The goal is to recognize which articles, which paragraphs, which items etc., a given mention refers to.

No.	Position Parts of Mentions	Context Information	Position of Antecedents (Which article, which paragraph, which items)
1	第九十条第一項第一号から第三号まで From the 1st item to the 3rd item of paragraph 1 of article 90	No need	--- *Referent 1* ----- (90, 1, 1) --- *Referent 2* ----- (90, 1, 2) --- *Referent 3* ----- (90, 1, 3)
2	同項 The same paragraph	-- *The current paragraph* -- 第九十六条第四項 article 96, paragraph 4	--- *Referent 1* ----- (96, 4)
3	前条第一項 The previous article, paragraph 1	-- *The current article* -- 第六条 article 6	--- *Referent 1* ----- (5, 1)
4	同項 The same paragraph	-- *The previous paragraph* -- 前条第一項 the previous article, paragraph 1	--- *Referent 1* ----- (5, 1)

Figure 4.8: Some examples of the output of the position recognition step.

We use regular expressions which are carefully constructed to recognize and determine the scope of antecedent candidates for a given mention. Some examples of position recognition results are given in Figure 4.8. The patterns of the position parts were described in Section 4.2.

Recognizing the position of the antecedent is usually simple when the position part of a mention is a complete one (i.e. the first example in Fig. 4.8). A position is complete if it

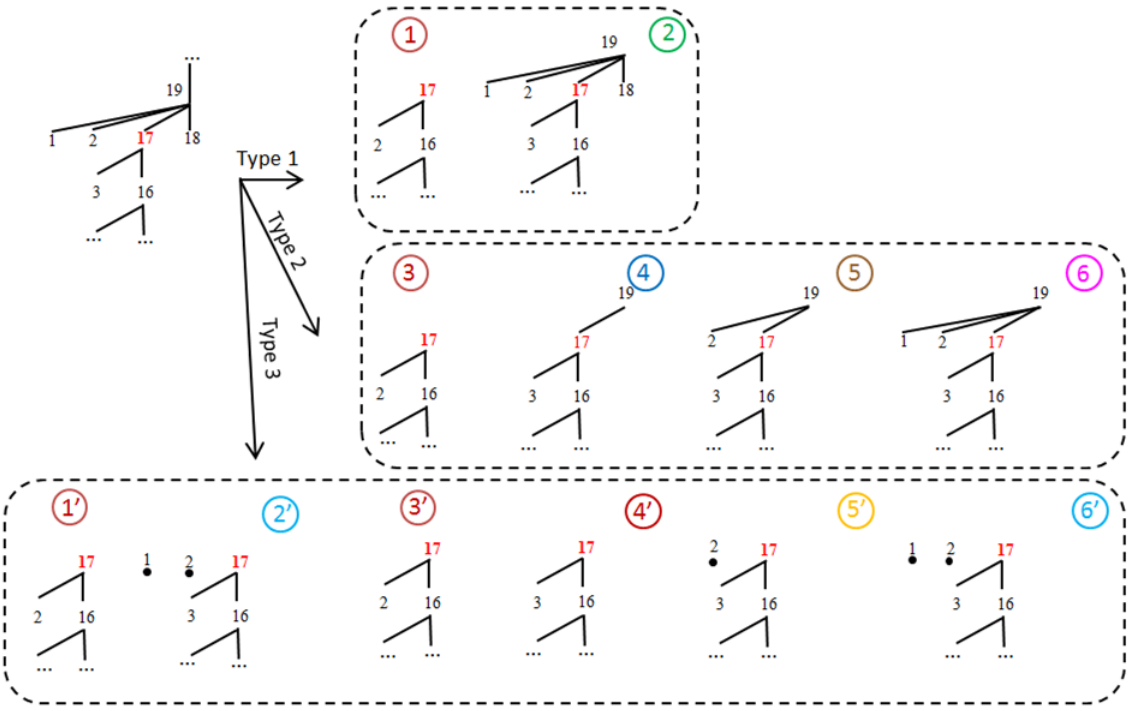

Figure 4.9: An example of generating candidates using strategy 1(a)($n_{head} = 17$).

completely includes the information of the document it refers to. We can find the article in the list. From this article, we will specify the precise location of the antecedent. Using regular expressions, we know where it should be located. For this type, we also use regular expressions to recognize complex positions and then resolve each of their positions.

Another group of positions that is a little bit harder to resolve is anaphora. We would like to list some typical cases as follows:

- References that refer to the current text: For example, this article, this paragraph, and so on. Recognizing this type of anaphora is feasible to resolve if we know the current location of the reference (i.e. the second example in Fig. 4.8).
- References that refer to an earlier point in the text, such as *the previous article*. These can be resolved using structure information of the law (i.e. the third example in Fig. 4.8).
- References that refer to an earlier reference, i.e. *dou kou (the same paragraph)*. This reference refers not to the current paragraph, but to a paragraph that was previously mentioned in the text (i.e. the fourth example in Fig. 4.8). It is usually the most recent documents referenced before and near the current reference in the text. To resolve this type of references, we need to keep a history of the references found so far.

4.5.4 Antecedent candidate generation

This step generates a list of candidates in the search space for a given mention. The search space is the output of the position recognition sub-step. We have two strategies

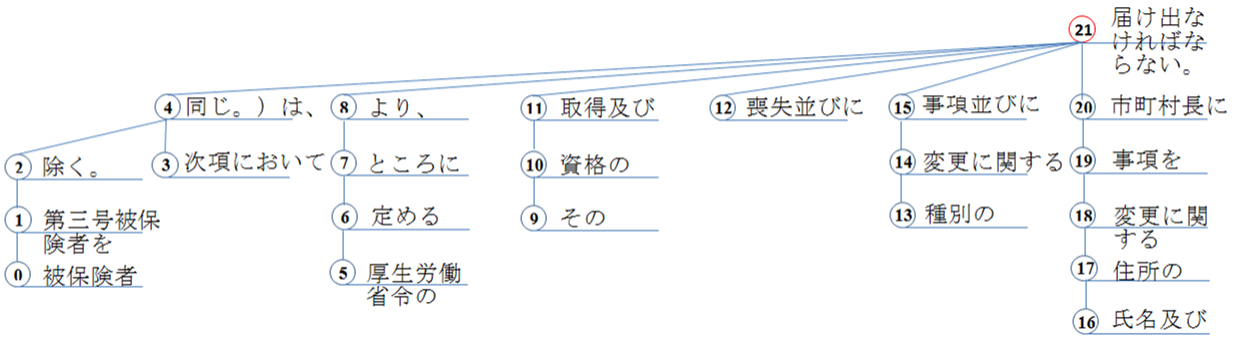

Figure 4.10: An example of parsing the sentence in the document of *Article 12, Paragraph 1.*

to generate candidates based on whether the mention head[7] (or synonyms of the mention head) of a mention appears in the antecedent sentences.

1. If the mention head or its synonyms appear, we generate candidates according to two ways:

   (a) Based on the dependency trees: We assume that the node containing the mention head or its synonyms is n_{head}.

   - Type 1 - Full subtrees: With each ancestor node of n_{head}, we extract all its descendants as a candidate.
   - Type 2 - Left subtrees: At each ancestor node of n_{head}, starting from the ancestor node's child that contains the node n_{head}, we gradually scan to the leftmost child of the ancestor node. At each currently scanned child, we extract all descendants of all the scanned children as a candidate. This originates from the fact that, in Japanese, an explanation of a noun usually appears on its left side.
   - Type 3 - We limit the boundary of all candidates of Type 1 and Type 2 which must be ended after the position of the mention head or its synonyms.

   (b) Based on the punctuation marks (usually a comma): With each candidate sentence, we perform a right-to-left scan from the position of the mention head or its synonyms. Each time we meet a comma or the beginning of a sentence, we will extract the scanned texts as a candidate.

   Figure 4.9 illustrates an example of generated candidates using strategy 1(a). Type 1 generates Tree 1 and Tree 2; Type 2 generates Tree 3, 4, 5, and 6; Type 3 generates Trees 1', 2' from Trees 1, 2 and Trees 3', 4', 5', and 6' from Trees 3, 4, 5, and 6. After that we remove trees that are duplicated from previously generated trees (Trees 1', 3, 3' and 4 are identical to Tree 1; Tree 6' is identical to Tree 2')[8] to create unique candidates.

[7] A mention head is the main noun of a mention. It identifies the intellectual entity that this mention contains.

[8] These identical trees are marked with the same color in Figure 4.9.

No.	Candidates	Nodes	Types
1	被保険者（第三号被保険者を除く。次項において同じ。）は、厚生労働省令の定めるところにより、その資格の取得及び喪失並びに種別の変更に関する事項並びに氏名及び住所の変更に関する事項を市町村長に届け出なければならない。 Pursuant to the provisions of the Ministry of Health, Labour and Welfare, the insured person (except for third type. The same for the next para),must notify to the mayor of a municipality matters relating to the change of name, address and matters relating to the change of type or loss and acquisition of the qualification.	all	Type 1
2	氏名及び住所の変更に関する事項を市町村長に届け出なければならない。 must notify to the mayor of a municipality matters relating to the change of name, address.	16-21	Type 2
3	種別の変更に関する事項並びに氏名及び住所の変更に関する事項を市町村長に届け出なければならない。 must notify to the mayor of a municipality matters relating to the change of name, address and matters relating to the change of type.	13-21	
4	喪失並びに種別の変更に関する事項並びに氏名及び住所の変更に関する事項を市町村長に届け出なければならない。 must notify to the mayor of a municipality matters relating to the change of name, address and matters relating to the change of type or loss.	12-21	
5	その資格の取得及び喪失並びに種別の変更に関する事項並びに氏名及び住所の変更に関する事項を市町村長に届け出なければならない。 must notify to the mayor of a municipality matters relating to the change of name, address and matters relating to the change of type or loss and acquisition of the qualification. (*correct case*)	9-21	
6	厚生労働省令の定めるところにより、その資格の取得及び喪失並びに種別の変更に関する事項並びに氏名及び住所の変更に関する事項を市町村長に届け出なければならない。 Pursuant to the provisions of the Ministry of Health, Labour and Welfare, must notify to the mayor of a municipality matters relating to the change of name, address and matters relating to the change of type or loss and acquisition of the qualification.	5-21	
7	*The same as candidate 1*	All	
8–14	*Established from candidates 1 to 7 by removing the word "なければならない" (means must)*		Type 3

Figure 4.11: Candidates generated by using the first strategy to generate candidates for the reference '*the notification in the provision of para 1*'.

To understand more about the first strategy of candidate generation, let us use the texts in *Article 12, paragraph 1* (see the texts in Section 4.1) as an example. We also wish to find the candidates for the reference '*the notification in the provision of para 1*' in *Article 12, paragraph 4*. The parse tree of the sentence in this document is presented in Figure 4.10. In this example, the node that contains the mention head '*notification*' is the node 21 (marked red), which has no ancestor node. By using strategy 1a), we generate 14 candidates as shown in Figure 4.11. Type 1 (full subtrees) yields one candidate (the first one). Type 2 (left sub-strees) yields the candidates from 2 to 7, in which candidate 5 is a true referent. Type 3 yields the candidates from 8 to 14. After that, we remove duplicated candidates, i.e. the candidates 1 and 7; and accordingly the candidates 8 and 14[9]. In all, we obtained 12 candidates for the given reference.

[9]These two candidates are created from the candidates 1 and 7 respectively.

2. If neither the mention head nor its synonyms appear, we obtain candidates for a mention as described in *1(b)*. However, the scan point starts from the end to the beginning of a candidate sentence.

4.5.5 Antecedent determination

This step aims at determining the exact antecedent for each mention. Depending on the classification of a mention, we have two strategies to determine its exact antecedent.

- For a mention of *Class 1*, using the information about the position of the antecedent, we can exactly determine the antecedent for this mention.
- For a mention of *Class 2*, from its candidates, we use a state-of-the-art model on the coreference resolution task to rank candidates based on the probability that the candidate is the antecedent of a given mention. The 1^{st}-ranked candidate will be selected as the exact antecedent.

To determine the exact antecedent for an active mention from a list of candidates, many models have been proposed, including mention-pair models [77, 91, 104], entity-mention models [62, 126], mention-ranking models [125] and cluster-ranking models [95]. Among these models, mention-ranking and cluster-ranking models are more robust than the first two models in the sense that they can address the limitation of previous ones [76, 95, 125]. These ranking models yield a theoretically more adequate and empirically better performance on some public standard corpora of the coreference resolution task. They directly capture the ranking on all the antecedent candidates of an active mention, instead of considering them independently. In this work, therefore, we choose a method of mention-ranking models to solve the task.

To train a ranker, we used the SVM ranker-learning algorithm SVM^{rank} [46] as the learning method. In the training phase, each instance $\mathbf{i}(m_i, c_j)$ is created based on a mention m_i and an antecedent candidate c_j. The feature sets used to represent each training instance are grouped into three sets which are described in more detail in Table 4.1. These linguistic features are supposed to be effective in estimating the probability that a candidate is an antecedent of an active mention. These feature sets are extracted using the n-gram information, the mention head (or its synonyms) and the dependency tree of the antecedent sentence. The first feature set tries to capture the boundary of the candidate c_j. The second feature set checks if the candidate c_j contains the mention head of m_i and calculates the distance from the position of the mention head to the beginning and the end of c_j. The third feature set captures how good c_j is a meaningful grammar unit as well as the smallest fragment of texts that supports the understanding of the mention m_i.

The way of assigning class values to each training instance is as follows: Assuming that S_k is the set of training instances created for a mention m_i, the class value for an instance $\mathbf{i}(m_i, c_j)$ in S_k is the rank of m_i among competing candidates as determined using the OFFSET [10] value.

[10]Why do we need the OFFSET value? Because generating the candidate that is the gold antecedent of a mention is a quite difficult task. Consequently, in this step, the system is unable to find the correct antecedent in some special cases. Moreover, the purpose of resolving mentions in legal texts is to show the referenced texts so that readers can quickly understand more about the rules that they are reading. Therefore, can we loosen the criteria to estimate whether the

Table 4.1: Feature sets extracted for the training instance $\mathbf{i}(m_i, c_j)$ ($position_{head}$: the position of the mention head in the antecedent sentence; $n_{meeting}$: the meeting node where the concatenation of all of its descendants covers the candidate c_j).

No.	Types	Feature names	Notes
1	N-gram	Start[1]	1^{st} word of c_j
		Start[1-2]	1^{st} and 2^{nd} words of c_j
		Start[1-3]	1^{st}, 2^{nd}, and 3^{rd} words of c_j
		End[1]	Last word of c_j
		End[1-2]	Two last words of c_j
		End[1-3]	Three last words of c_j
2	Mention Head	ContainMentionHead	Does c_j contain the mention head of m_i?
		DistanceToBeginning	The distance from $position_{head}$ to the beginning of c_j
		DistanceToEnding	The distance from $position_{head}$ to the end of c_j
		MentionHeadAtBegin	Does m_i appear at the beginning of c_j?
		MentionHeadAtEnd	Does m_i appear at the end of c_j?
3	Dependency	CountNodesToMeetingNode	The distance from the node containing the mention head to the meeting node $n_{meeting}$
		NumberOfLeftChilds	At node $n_{meeting}$, count the number of children from its leftmost child to the child that contains the first word of the candidate c_j
		NumberOfRightChilds	At node $n_{meeting}$, count the number of children from its rightmost child to the child that contains the last word of the candidate c_j

$$value\ \boldsymbol{i}(m_i, c_j) = \begin{cases} 1 & \textit{if } c_j \textit{ is the gold antecedent of the mention } m_i \\ 0.5 & \textit{if } c_j \textit{ covers the gold, and the number of words outside the gold is less than OFFSET} \\ 0 & \textit{otherwise} \end{cases}$$

After training, this mention-ranking model will be used to rank the candidates for a given mention. We create test instances for a mention by pairing the latter with each of its antecedent candidates. In the testing phase, the candidate that is assigned the largest value by the ranker is selected as the exact antecedent of that mention.

4.6 Experiments

4.6.1 Corpus

The Japanese National Pension Law (JNPL) corpus on reference resolution was manually built by law-making experts. In this corpus, all references that refer to the inside scope of the JNPL were marked. However, the JNPL does not include the references that refer to other laws, or refer to ambiguous ranges. The architecture of the JNPL is shown in Figure 4.12. The law consists of articles, articles consist of paragraphs, and paragraphs

output of the system is considered to be the same as the gold antecedent? Instead of exactly matching, we allow the output to exceed the boundary of the gold antecedent with the smallest number of words possible. If the output contains the gold antecedent, and the total number of words in additional texts is not greater than the OFFSET value, the output is considered to be true. In experiments, we set OFFSET to be equal to 10.

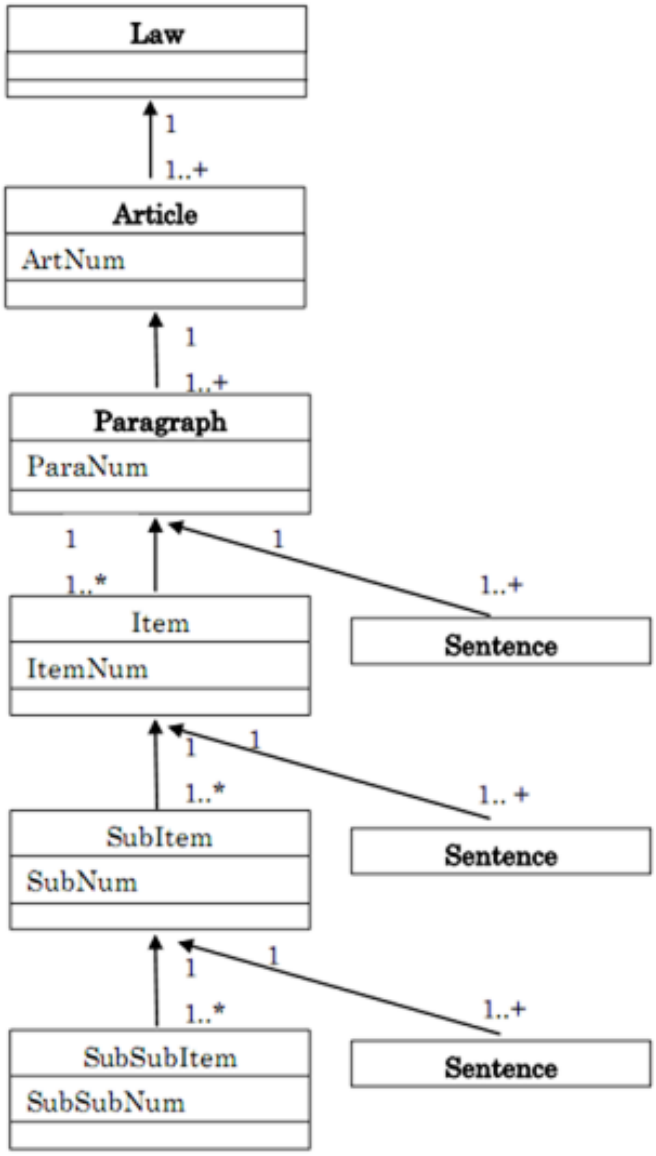

Figure 4.12: The architecture of the JNPL corpus on reference resolution.

consist of sentences. A sentence may belong to items, sub-items, or sub-sub-items of a paragraph. Below are some statistics about this corpus:

- # Articles: 99 articles
- # Paragraphs: 931 paragraphs
- # Mentions: 748 mentions (586 mentions of Class 1, and 162 mentions of Class 2). One mention can refer to one or more referents.

4.6.2 Experimental setup

We divided the JNPL corpus into 10 sets, and conducted 10-fold cross-validation tests for all experiments. For the mention detection task which we modeled as a sequence labeling

problem, we evaluated the performance of our system by precision, recall, and the F1 score as follows:

$$P = \frac{\#correctly\ detected\ mentions}{\#detected\ mentions}, R = \frac{\#correctly\ detected\ mentions}{\#gold\ mentions},$$
$$F1 = \frac{2*Precision*Recall}{Precision+Recall}.$$

For the remaining four tasks (the mention splitting, the mention classification, the position recognition and the antecedent determination steps), we conduct experiments using the gold mentions of the corpus. With each mention, the system always outputs results for its input. Therefore, to estimate the performance of these sub-systems, we use the accuracy score to evaluate the experimental results.

$$Accuracy = \frac{\#correctly\ processed\ mentions}{\#gold\ mentions}.$$

where the numerator, *#correctly processed mentions*, corresponds to the number of mentions that are correctly split in the mention splitting task; correctly classified in the mention classification task; correctly position-recognized in the position recognition step and correctly antecedent-determined in the antecedent determination step.

To estimate the performance of the whole system, we use precision, recall and the F1 score as follows:

$$P = \frac{\#correctly\ detected\&resolved\ mentions}{\#detected\ mentions},$$
$$R = \frac{\#correctly\ detected\&resolved\ mentions}{\#gold\ mentions}, F1 = \frac{2*Precision*Recall}{Precision+Recall}.$$

where the *#correctly detected&resolved mentions* refers to the total number of mentions that are correctly detected and correctly antecedent-determined.

4.6.3 Experimental results

Mention detection and mention splitting

To learn the models, we used the CRF++[11] tool written by Kudo. In extracting feature sets, we used a combination of n-gram (n $\leq$ 3) of words, part-of-speech tags, and chunking information. This information was taken from the output of the Cabocha[12]. Cabocha [53] is a Japanese dependency structure analyzer based on Support Vector Machines (SVMs) [115]. When performing experiments, we used three kinds of label settings which are the IOB, IOE, and FIL notations [60]. We also investigated the task using different combinations of feature sets to determine which feature sets yield better performances.

Table 4.2: Experimental results for the mention detection task (%).

Notations	Word			Word + POS			Word+POS+Chunk		
	R	P	**F1**	R	P	**F1**	R	P	**F1**
IOB	79.05	76.95	77.95	80.6	79.32	79.92	80.58	79.6	80.06
IOE	79.2	77.93	78.53	79.63	78.3	78.92	79.99	79.09	79.51
FIL	79.88	77.83	**78.82**	80.51	79.49	**79.95**	80.7	79.66	**80.14**

[11] http://crfpp.googlecode.com/svn/trunk/doc/index.html
[12] http://code.google.com/p/cabocha/

Table 4.3: Experimental results for the mention splitting sub-step (Accuracy (%)).

Notations	Word	Word + POS	Word+POS+Chunk
IOB	99.23	99.17	**99.3**
IOE	99.17	99.17	**99.3**
FIL	99.17	99.17	**99.3**

Table 4.4: Experimental results of the mention classification task (Accuracy (%)).

	Content Words	**All Words**
Baseline	78.34	78.34
MEM	87.03	86.54
SVM	**87.16**	**87.23**

Experimental results of the mention detection sub-step are shown in Table 4.2. We realize that the more feature sets we use, the better the results are. This is reasonable because these feature sets yield more benefits to the model. We obtained the highest performance of 80.14% in the F1 score using the FIL notation on feature sets which are based on words, part-of-speech tags and chunking information. That is because most mentions begin with some words clearly locating positions such as *dai*, *tsugi*, *mae*, *dou*, etc; and end with a noun phrase. In some cases, the NPs were nested, which slightly worsened the performance of the IOE notation in comparison with that of the IOB notation. In all settings, the recall was higher than the precision score. This can be attributed to the fact that some detected mentions refer to the outside scope of the corpus and therefore are not annotated.

Experimental results of the mention splitting sub-step are given in Table 4.3. For all three notations, we obtained a very high accuracy of 99.3% when using words, part-of-speech tags, and chunking features. The reason for this high performance is that most mentions have their position parts ending with particles like *ni* or *no*.

Mention classification

To perform this step using MEM, we used the public MaxEnt classifier[13] tool with the BLMVM optimization algorithm [10]. We also used an SVM implementation of LibSVM[14] of [20], which is an integrated software for support vector classification. As an appropriate baseline model, we chose the model that assigns all mentions to the larger class (*Class 1* by default).

Experimental results in Table 4.4 show that using only the content words results in a higher accuracy than using all words of a mention with the two machine learning methods. This is because position words do not usually convey the information about the class of a mention. For this step, SVM performs slightly better than MEM on both settings. The results also show that our system outperforms the baseline model by ≈8% in terms of accuracy.

[13]Downloaded from: http://www-tsujii.is.s.u-tokyo.ac.jp/~tsuruoka/maxent/

[14]Downloaded from http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Position recognition

By using regular expressions and context information, we were able to locate up to 96.18% of correct positions of all mentions. The few cases where our system was unable to recognize the correct position can be classified under some exceptions that do not conform to the regular expressions that were described in Section 4.2.

Antecedent determination

Table 4.5: Experimental results of the antecedent determination step.

No.	Types of features	Accuracy (%)
1	n-gram	84.76
2	n-gram + Head	85.40
3	n-gram + Head + Dependency	**85.61**

In this chapter, we chose to run an algorithm proposed by [95]. To train a ranker, we used as the learning method the SVM ranker-learning algorithm SVM^{rank} [46] [15], an instance of SVM^{struct} for efficiently training Ranking SVMs [45].

Table 4.5 presents the experimental results of determining a true antecedent among candidates for each mention in the corpus. We conducted experiments by adding feature sets incrementally to observe their effect on the model. The experimental results indicated that the combination of feature sets of n-gram, the mention head and the dependency information yields the best performance. We obtained 85.61% accuracy in determining the antecedents for all mentions in the JNPL corpus. This table also shows that we achieved better performance if we integrate more feature sets into the model.

This section also measures whether the output contains the gold antecedent and the length of additional texts in terms of words is greater than the OFFSET value, but not the entire document. The accuracy is 90.63%.

In this step, we also conducted experiments to compare two approaches to this problem as discussed in the previous chapter: the pairwise and the listwise approaches. Table 4.6 shows the experimental results of these two approaches using three types of features.

Table 4.6: Experimental results of the antecedent determination step using two approaches: the pairwise and the listwise.

Approaches	Methods	Accuracy (%)
Pairwise	SVMRanking	85.61
Listwise	ListNet	85.3
	ListMLE	**86.03**

We kept the learning rate at 0.01 and varied the tolerance rate and the number of iterations to see the effect of the listwise methods on this step. As we can see that, if we have a good development set, in the ListMLE method, we can determine several values for the tolerance rate, at where the accuracy is better than that of the SVMRanking method. However, in the ListNet method, we did not see any improvement by varying

[15] This software can be downloaded from http://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

the number of iterations. Figures 4.13 and 4.14 illustrate the accuracy curves of two listwise methods with different parameter values (the red line is the maximum accuracy using the SVMRanking method).

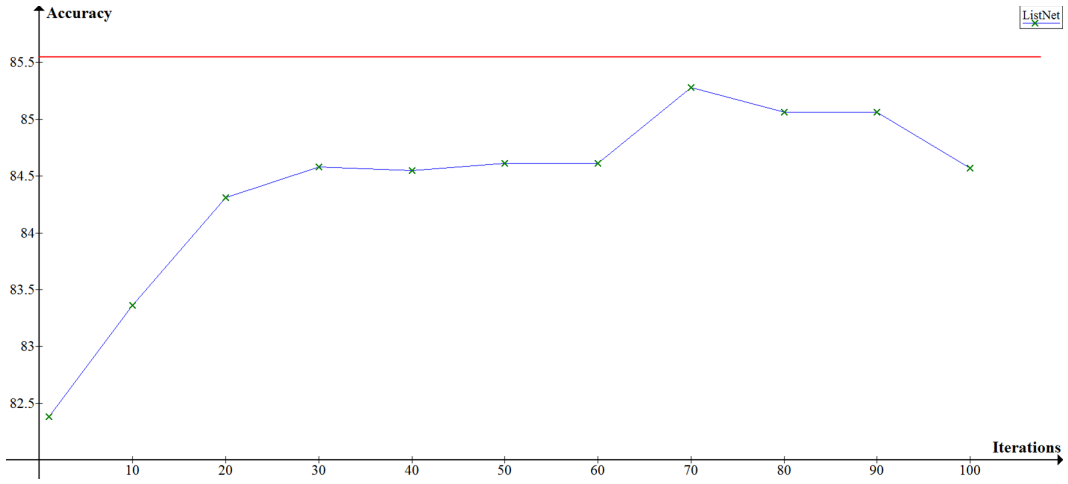

Figure 4.13: The accuracy of the ListNet method depends on the number of iterations (the learning rate is fixed at 0.01).

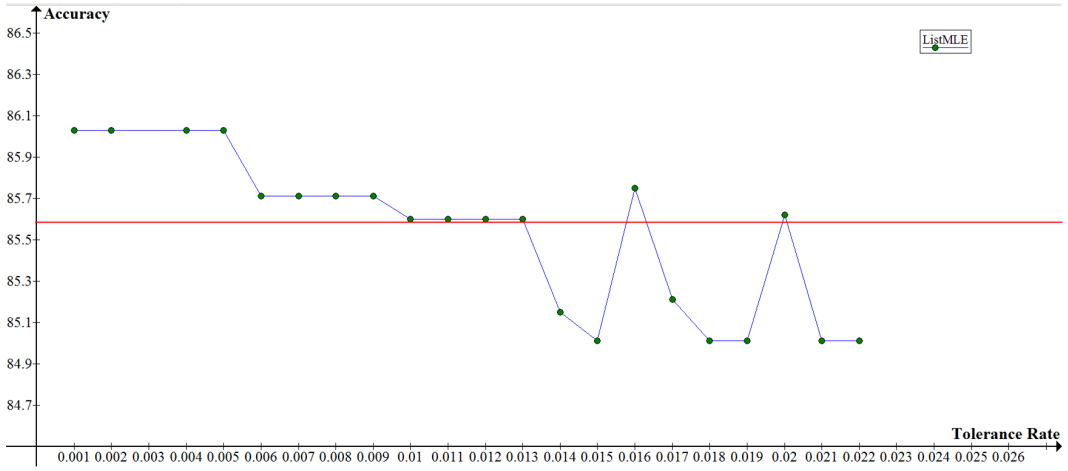

Figure 4.14: The accuracy of the ListMLE method depends on the tolerance rates (the learning rate is fixed at 0.01).

We also measure the performance of each class (namely Class 1 and Class 2) for this step. For the class 2, we got the accuracy of about 34%. The accuracy of Class 1 is perfect because in this step we use the gold data.

4.6.4 Analyzing the impact of each step on the final system

Table 4.7 shows the experimental results of analyzing the impact of each step on the final system based on the following assumptions:

- Experiment 1: assumes that the mention detection step correctly detects all mentions in the corpus. If that occurred, the whole system achieved 79.23% in the F1 score.

Table 4.7: Experimental results of the effect of each step on the final system (MD, MS, MC, and PR stand for the Mention Detection, Mention Splitting, Mention Classification and Position Recognition steps respectively).

No.	Assumptions	Precision	Recall	F1
1	Step 0 (MD) is true	79.73	78.74	79.23
2	Step 1.1 (MS) is true	67.42	66.65	67.03
3	Step 1.2 (MC) is true	71.53	70.71	71.12
4	Step 1.3 (PR) is true	67.48	66.66	67.07
5	Step 1 (MC+MS+PR) is true	71.56	70.72	71.14
6	Step 0 and Step 1 are true	86.74	85.61	86.17
7	End-to-end system	67.42	66.63	67.02

- Experiments 2,3, and 4: assume that the mention splitting, the mention classification and the position recognition sub-steps perform perfectly respectively.
- Experiment 5: assumes that Step 1 - Contextual Information Extraction - is correctly performed. With that assumption, the whole system achieved 71.14% in the F1 score.
- Experiment 6: assumes that the first two steps perform perfectly. If that occurred, we obtained 86.17% in the F1 score for the whole system.
- Experiment 7: does not assume that any step perform perfectly, and the performance reached 67.02% in the F1 score.

To comprehend the impact of each step on the whole system, it should be noted that the higher the F1 score, the more important the step is. From these results, we can see that the mention detection step plays the most important role in the performance of the whole system. The next important step is the contextual information extraction. Since this step includes 3 sub-steps, the order of the role of each step in importance in the final system is as follows: the mention classification sub-step, the position recognition sub-step, and the mention splitting sub-step.

The last experiment is also the end-to-end system in which the input of the current step will be the output of the previous step. In the whole task setting, we obtained a result of 66.63% recall, 67.42% precision, and 67.02% in the F1 score in determining antecedents for mentions needed to be resolved in the corpus. This performance is quite promising.

4.6.5 Improving the performance of the final system

This section looks at the limitation of the proposed four-step framework and suggests a solution for overcoming this limitation.

Limitation of the framework

We proposed a cascade framework, where the output of the previous step was used as the input of the next step. So, the errors of the previous step could be propagated to the next step. By analyzing the effect of each step on the final system, we saw that the

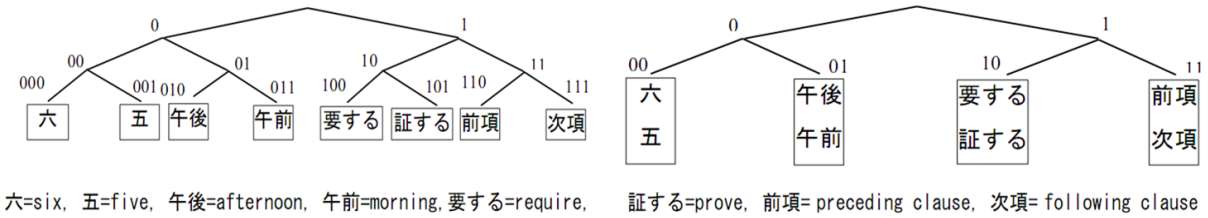

Figure 4.15: An example of Brown word-cluster hierarchy.

first step - mention detection has the biggest influence on the performance of the whole system. Experimental results of each step showed that this step along with the mention classification step yield the lowest performance. Hence, in this section, we would like to focus on improving these two phases through the consideration of using the Brown clustering algorithm to improve the performance.

A simple semi-supervised method using Brown clustering

This sub-section will briefly describe the Brown clustering algorithm and show how to use the Brown word clusters to improve the mention detection step and the mention classification sub-step. Among the word representation methods, we chose the Brown clustering algorithm for our work because of its simplicity and efficiency.

The Brown clustering algorithm is a word clustering algorithm based on the mutual information of bigrams [15]. The input to the algorithm is a set of words and a text corpus. In the initial step, each word belongs to its own individual cluster. The algorithm then gradually groups clusters to build a hierarchical clustering of words.

Figure 4.15 shows an example of a Brown word-cluster hierarchy in a binary tree style. In this tree, each leaf node corresponds to a word, which is uniquely identified by its path from the root node. This path can be represented by a bit string, as shown on the left side. From the root node, we add bit 0 to the left branch and bit 1 to the right branch. A word-cluster hierarchy is reduced to a depth of n if all words with the same n-bit prefix are grouped in one cluster. For example, if the word-cluster hierarchy on the left side of Figure 4.15 is reduced to a depth of 2, we will obtain a new hierarchy on the right side of Figure 4.15.

Features extracted at n-bit depth are binary strings of length n. By reducing the word-cluster tree to different values of depth n, we can group words at various levels, from coarse clusters (small value of n) to fine clusters (large value of n). These features are used as extra word features. Next, we will present how to integrate into the models in order to improve the performance of the mention detection step and the mention classification sub-step.

Mention detection step with extra word features

The main idea of our semi-supervised learning method is to use *unsupervised* word representations (Brown word clusters) as extra word features of a *supervised* model. In this framework, *unlabeled data* is used to produce word clusters. From these word clusters, we extract extra word features, and add these features to a *supervised* model (*labeled data* are used to train this model). Figure 4.16 shows our semi-supervised learning framework. This framework consists of two phase: the *unsupervised* phase with the Brown clustering algorithm, and the *supervised* phase with CRFs.

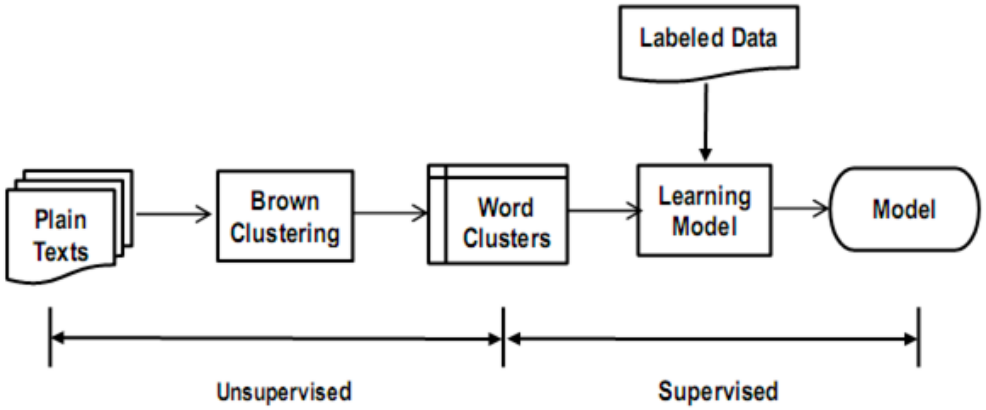

Figure 4.16: Semi-supervised learning framework.

Table 4.8: Mention Detection: Experimental results when integrating extra word features using Brown Clustering information (- means that we did not use extra word features, + means that we used extra word features).

No.	Label Settings	Features	Precision	Recall	F1
1	IOB	-	80.58	79.60	80.06
		+	**80.97**	**80.13**	**80.51**
2	IOE	-	79.99	79.09	79.51
		+	**80.82**	**80.9**	**80.82**
3	FIL	-	80.70	79.66	80.14
		+	**80.55**	**81.01**	**80.74**

To produce word representations, we first collected plain text from the Japanese law translation website[16]. This website provides many Japanese law articles in both Japanese and English. The plain text corpus which we downloaded included more than 67.000 sentences of Japanese laws. We first performed word segmenting using Cabocha tool, and then conducted the Brown clustering algorithm to cluster words. We used the implementation of Percy Liang [58], and set the number of clusters at 200.

For the mention detection task, we extracted features at 4-bit depth and 6-bit depth. We integrated these features into three models which are the best models of each notation. The results in Table 4.8 show that using extra words features improves the models on all three kinds of notations. For the best performance, the system increases the F1-score from 80.05% upto 80.82%.

Mention classification step with extra word features

With the same methodology when integrating extra word features of the mention detection step, we also conducted experiments for the mention classification sub-step. We used the same setting when extracting extra word features. The experimental results of the semi-supervised method with extra word features are shown in Table 4.9. The results showed that using extra words features improves the models on the two learning algorithms, which are SVM and MEM. For the best performance, the system increases the F1-score from 87.03% up to 87.86% on MEM, and fro 87.16% up to 87.88% on SVM.

[16]http://www.japaneselawtranslation.go.jp

Table 4.9: Mention Classification: Experimental results when integrating extra word features using Brown Clustering information.

Classifiers	Features	Content Words	All Words
MEM	-	87.03	86.54
	+	**87.86**	**87.08**
SVM	-	87.16	87.23
	+	**87.88**	**87.40**

4.6.6 A true working example of using the final system

Figure 4.17 gives an example of the systems output. It shows the detail process of how the system works. The input is a piece of an article which is in the form of a sequence of characters. This input contains a mention *'The employer who employs the second type of insured person provided in the preceding paragraph'* that should be detected. The first mention detection step correctly recognizes this mention and lets this mention go through Step 1. In this step, Sub-step 1.1 splits it into two parts which are a position part (*'in the preceding paragraph'*) and a content part (*'The employer who employs the second type of insured person provided'*). Based on the content part, Sub-step 1.2 classifies this mention to Class 2. This means that this mention only refers to a fragment of a document. Based on the position part, Sub-step 1.3 recognizes the position that contains the referenced text fragment of this mention, which is in *Article 12, Paragraph 6*. The outputs from Sub-step 1.2 and 1.3 are then used in combination to generate all possible antecedent candidates for this mention through Step 2. The last step - Antecedent Determination - determines exactly the true referenced texts (the texts in green) for this mention among its candidates.

4.7 Error analysis

This research focused on two major problems. The first one is to detect all references (mentions) in the legal texts, and the second one is to resolve these mentions to their correct referenced expressions (antecedents). This section analyzes error cases and reasons that contributed to the instance of failures in our system.

By observing the output of the final system, we realized that in detecting mentions, most of errors can be attributed to the following reasons:

- Detected but they are not labeled as mentions in the corpus (Because, these mentions refer to the documents which is beyond the scope of JNPL corpus (i.e. the first example in Figure 4.18)).
- Mentions are detected beyond the true boundary as labeled in the corpus. This is usually caused by ambiguities of mentions, in which the content parts are usually nested noun phrases (i.e. the second example in Figure 4.18)).

In determining antecedents, most of errors are caused by the following reasons:

- Mentions are wrongly split, classified, and position-recognized in intermediate sub-steps.

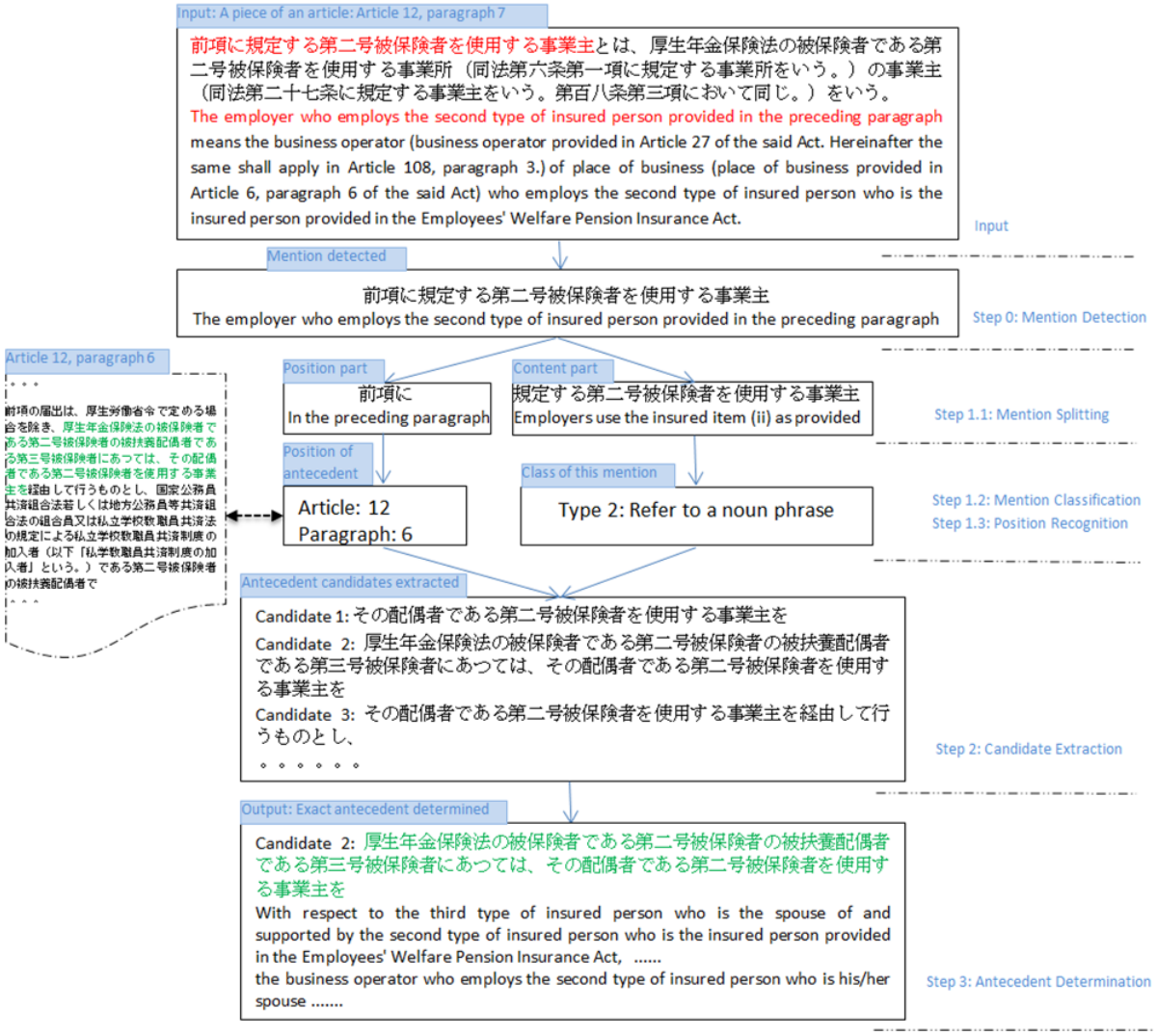

Figure 4.17: An output example of our system.

- The antecedent candidate generation step did not get out the gold antecedent for mentions. Among all mentions that we considered, there are approximately 90% of mentions whose generated candidates contains their gold antecedents. Therefore, it is impossible to find out the gold antecedents for the remaining 10% of mentions.
- Dependency parsers typically could not deal well with long sentences in the legal domain. So, the information extracted using dependency trees is not highly reliable in some cases.

4.8 Discussion

In this section, we discuss more our system. We first discuss the performance of the system in comparison to other related systems. Then, we discuss the versioning problem of our work.

No.	Error Examples
1	この法律において、「保険料納付済期間」とは、第七条第一項第一号に規定する被保険者としての被保険者期間のうち納付された保険料（第九十六条の規定により徴収された保険料を含み、第九十条の二第一項から第三項までの規定によりその一部の額につき納付することを。。。
2	"第一項の指定の手続その他前三項の規定の実施に関し必要な事項は、厚生労働省令で定める。"

Figure 4.18: Some error examples of the mention detection step.

4.8.1 Comparison with previous work

Previous sections (Section 4.1 and Section 4.2) have distinguished the difference of this work and previous related work. References which are considered in this research consist of two main parts, i.e. position parts and content parts. This form is slightly different from the form of normative references, which are studied in previous work. Normative references only consist of position parts, which are used to determine the location of referenced documents. References in this research are extended to cover not only position parts but also content parts. These content parts are used to determine the smallest fragments of texts in the referenced documents, which are referred to. This is the novel point of this research. Compared with previous work, therefore, our system yields a lower performance[17]. This is understandable because our system aimed at a more complete target, which tries to resolve references to the smallest fragment of texts rather than only locating the referenced document[18]. This resolution requires deep syntactic and semantic knowledge of the texts. Hence, detecting and resolving this kind of references is much more difficult in comparison with normative references in previous work.

4.8.2 The versioning problem of laws

In this research, we did not take versions of laws into account. This means that our corpus only contains one version of the JNPL law. If the law changes[19], the system should be changed a little bit to adapt to that change. In this case, the information about how laws are changed will be used to help the resolver to extract the expected referents (extract only old referents, or only new referents, or both depending on the real applications). This interesting problem will be considered in future work.

4.9 Conclusion and future work

In this chapter, we presented a new interesting task of reference resolution in legal texts. The goal is to create a system which can automatically extract referents for references in real time. In this research, we go a step further in comparison to previous work by extracting the smallest fragments of texts in the referenced documents. This does not only help readers in understanding the law, support law-makers in developing and amending laws, but also support in building an information system which works based on laws, etc.

Based on the characteristics of reference phenomena in legal texts, we proposed a framework with four steps to solve the task. In comparison to reference resolution in

[17]Previous work reported F scores of more than 90% for normative reference detection and resolution

[18]In other research, we studied on detecting and resolving normative references. The experimental results showed that we also obtained more than 90% in the F1 score on the same JNPL corpus on normative references.

[19]The provision of a document A is changed, and therefore A is changed to A'.

general texts, our framework consists of two extra steps (Step 1 and 2), which take advantage of the reference phenomena in legal texts to help solve the task more effectively. Experiments on the JNPL showed very promising results. We achieved a performance of 80.06% in the F1 score in the mention detection step; a 99.3% accuracy in splitting mentions into two parts; a 96.18% accuracy in locating the positions of antecedents; an 87.03% accuracy in classifying them. In the final antecedent determination step, the system achieved 85.61% accuracy in determining the antecedents for all mentions in the corpus. Our final end-to-end system achieved 67.02% in the F1 score in the whole task of detecting and determining antecedents for mentions.

There is a wide range of open possibilities for improving this important task. For example, by conducting experiments to estimate the effects of each step on the final model, we see that an improvement of this framework should imply paying more attention to the mention detection step. In addition, we also should take care of solving error cases analyzed in the error analysis section. In future research, we will exploit the output of this system to find contradictions in legal texts. We will also build a QA system to support citizens in accessing legal documents.

Chapter 5

Answering Legal Questions by Mining Reference Information

This chapter presents a study on exploiting reference information to build a question answering system restricted to the legal domain. Most previous research focuses on answering legal questions whose answers can be found in one document[1] without using reference information. However, there are many legal questions whose answers could not be found without linking information from multiple documents. This connection is represented by explicit or implicit references. To the best of our knowledge, this type of questions is not adequately considered in previous work. To cope with them, we propose a novel approach which allow us to exploit the reference information among legal documents to find answers. This approach also uses requisite-effectuation structures of legal sentences and some effective similarity measures to support finding correct answers without training data. The experimental results showed that the proposed method is quite effective and outperform a traditional QA method, which does not use reference information.

5.1 Introduction

A question answering (QA) system is a system that is able to automatically respond answers to questions posed by human in a natural language by retrieving information from a collection of documents. This is an important task and has drawn much attention in natural language processing research. Particularly, there are several top conferences which have organized special tracks for the topic of QA such as Text Retrieval Conference (TREC[2]) and Cross Language Evaluation Forum (CLEF[3]).

When considering an application of QAs in a specific domain, especially the legal domain, we saw that there is little work particularly devoted to this kind of research, despite its wide uses and applications. In the legal domain, QAs could be applied to help citizens and law-makers have easier access to legal information. Previous works [2, 28, 85] showed that a common problem is that traditional QAs are not adequate to find the correct answers to legal questions. This was mostly caused by special structures, specific terms and long sentences.

[1]The term '*documents*' corresponds to articles, paragraphs, items, or sub-items according to the naming rules used in the legal domain.

[2]http://trect.nist.gov

[3]http://www.clef-campaign.org

Question	市町村長は、被保険者の氏名及び住所の変更に関する事項の届出を受理したときは、何をしなければなりませんか？ "What should a mayor of a municipality do if he/she receives the notification on changing the address or name of an insured person?"
Clues	***Article 12, paragraph 1*** 被保険者（第三号被保険者を除く。次項において同じ。）は、厚生労働省令の定めるところにより、[*その資格の取得及び喪失並びに種別の変更に関する事項並びに氏名及び住所の変更に関する事項を市町村長に届け出なければならない*]。 Pursuant to the provisions of the Ministry of Health, Labour and Welfare, the insured person (except for the third type. The same for the next para), must [*notify to the mayor of a municipality matters relating to the change of name, address, as well as matters related to change of type and loss and acquisition of the qualification*]. 。。。。。。。。。。。。。。。。。。。。 ***Article 12, paragraph 4*** "市町村長は、<*第一項又は第二項の規定による届出*>を受理したときは、厚生労働省令の定めるところにより、厚生労働大臣にこれを報告しなければならない。" When a mayor of a municipality receives <*the notification in the provision of Para 1 or Para 2*>, pursuant to the provisions of the Ministry of Health, Labour and Welfare, s/he must report it to the Minister of Health, Labour and Welfare.
Answer	厚生労働省令の定めるところにより、厚生労働大臣にこれを報告しなければならない。 pursuant to the provisions of the Ministry of Health, Labour and Welfare, s/he must report it to the Minister of Health, Labour and Welfare.

Figure 5.1: A question is solved in this chapter. In this figure, references are bounded in angle brackets (⟨⟩) while their referents are bounded in square brackets ([ ]).).

In many laws, it is common that some specific terms are first defined in one document and then referred many times in other documents by using briefer expressions. Besides, when editing laws, lawmakers usually import the explanation or description of application conditions, purposes, provisions, etc. from other laws if available. This, as a result, helps to guarantee the soundness as well as the consistency in a law system. In order to link documents, references are used to identify the position as well as the fragment of texts referred to. These references bring precious information and their resolution helps interpreting the law. This is a useful characteristic of legal texts which can benefit the process of finding correct answers to many legal questions. Until now, however, there is no research on using this advantage of references to help finding correct answers. Of several work dedicated to legal QAs [2, 28, 85, 111], they mostly focus on legal questions whose answers can be found in only one document. However, the fact is that many legal questions requiring answers are combined from multiple documents which are linked based on references (as we can see later in Section 3). This type of questions is not adequately considered in previous research[2, 28, 85, 111]. Building a good reference resolver is not a trivial task. Therefore, for some languages, authors usually find an alternative solution to help indirectly representing the relationship between documents[69] rather than using the reference information.

In this chapter, we investigate an application of reference resolution to a QA system restricted to the legal domain. We focus on one type of questions which can be benefited

from the reference information such as the example in Figure 5.1. In this example, to answer the question, it is necessary to link the information from two documents to find the answer. The linking information is expressed via the relation of the reference-referent in the colored italic texts. The red texts bounded in angle brackets of the document '*Article 12, Paragraph 4*' is a reference, which refers to the green texts bounded in square brackets of the document '*Article 12, Paragraph 1*'. Such questions are quite popular in the legal domain.

Sometimes, users are not only interested in obtaining just an answer, but also want to know its evidence. In this chapter, therefore, we also give proofs of the answer. The main contribution of our work can be concluded in the following points:

- Building a QA system for Japanese legal texts, based on the reference information.
- Adequately considering one type of legal questions that can be benefited from reference information.
- Testing the proposed system on some legal questions yields promising performance, and outperforms a traditional QA system.

This chapter is organized as follows. Section 5.2 presents related work. Section 5.3 describes important characteristics of legal texts exploited in this research, i.e. references-referents structures between documents. In this section, we also describe in more details the type of legal questions considered in this work. Section 5.4 presents a proposed framework, which exploits the characteristics of legal texts, especially the reference information shown in Section 3. Section 5.5 presents experiments to compare the proposed system with a traditional QA system. Finally, Section 5.6 concludes the chapter and discusses future research.

5.2 Related Work

This section presents previous work closed to our work. We present two kinds of related work which are question answering using coreference information in general texts and in the domain of legal texts.

5.2.1 Question Answering using coreference information in general texts

Various research projects have investigated how coreference information can be employed to determine the contexts that contribute potentially relevant information about entities mentioned in a question(as relevant for QA). Most of them concluded that Question Answering is known to benefit from the availability of coreference information [13, 33, 39, 71, 72, 106, 120]. Coreference is necessary to resolve cases such as: *How much did Mercury spend on advertising in 1993?*. The sentence which contains the answer in *Last year the company spent 12m on advertising* and *the company* refers to *Mercury* three sentences earlier.

For details, Morton [71] attempts to find coreference relationships between the entities and events evoked by the query and those evoked in the document. Based on these

relationships, sentences are ranked, and the highest ranked sentences are displayed to the user. The coreference relationships that are modeled by this system include identity, part-whole, and synonymy relations. In other research [72], Morton also present a system which retrieves answers to queries based on term weighting supplemented by coreference relationships between entities in a document. An evaluation of this system is given which demonstrates that the coreference relationships allow significantly more questions to be answered than with a baseline system which doesnt model these relationships.

In [33], Gaizauskas et al., built a QA system in which coreference is applied to the snippets obtained from the search engine in order to obtain all the information available about the entities in the question.

In [13], Breck presented a QA system participated at TREC-8. In that research, they also showed that QA can be solved by employing coreference information in the two stages of (1) relating entities mentioned in the query to the retrieved doc-uments, and (2) looking at the relevant coreference classes and searching the contexts in which these entities occur for information that may contribute to answer the question.

Not only QA in English can be benifited from coreference information as in previous research, there also other research for non-English languages such as Ductch [39], etc.

Previous work in general texts showed that QA benefits from the availability of coreference information since it renders possible the identification of contexts in which information regarding the entities a question is about is contributed.

5.2.2 Question Answering using coreference information in legal texts

In the domain of legal texts, there is not much research dedicated to QAs.

In [85], Paulo et al. present a QA system for Portuguese juridical documents. The proposed approach is based on computational linguistic theories: syntactical analysis followed by semantic analysis; and finally, a semantic/pragmatic interpretation using ontologies and logical inference. The QA system was applied to the complete set of decisions from several Portuguese juridical institutions. It uses very expensive sources. The application texts legal domain is not law texts, therefore, it cannot use the characteristic of the law.

Monroy et al. [69] focus on building a QA system for Spanish at the shallow level by using graphs. The system gives answers which consist of a set of articles related to the question and also the relevant articles related with them to complement the answer. This method represents the link between documents by using the similarity (i.e. TF.IDF measure) between them via terms in documents. They also limit questions that mainly ask if it is possible to perform certain action or not.

Recently, Tomura et al. [111] present a study on building a QA system for Japanese legal texts. In this work, they deal with 5 types of questions whose answers can be found from one document using the requisite-effectuation structures of law sentences. This work shares the same type of law with our work - the Japanese National Pension Law (JNPL).

To the best of our knowledge, there exists no work on QA, which focuses directly on making use of reference information between legal documents.

5.3 A Type of Legal Questions Raised from Characteristics of Legal Texts

Firstly, we introduce some important characteristics of legal texts. Then, we describe a type of legal questions, which is mostly raised from these characteristics.

5.3.1 The Characteristics of Legal Texts

One important characteristics of legal texts is that they usually have some specific structures at both sentence and paragraph levels. At sentence levels, law sentences usually have some specific structures[5]. At paragraph levels, sentences in the same paragraph usually have close relations. Another important characteristic of legal texts is that, at the discourse level, legal documents contain many reference phenomena which need solving in order to understand their contents.

Reference phenomena in Legal Texts

Legal texts contain many reference phenomena within them. Legal references relate to terms, definitions, provisions, etc. For example, when law-makers describe conditions of a law in *Article 12, Paragraph 4* of the JNPL, they recall the definition of a type of notification by using a reference *'the notification in the provisions of Para 1 or Para 2'*. If this reference is resolved, we can fully understand which notification (explained in *Article 12, Paragraph 1*) is actually referred to in this document.

References (Mentions) [114] in legal texts have their own structures, which are different from mentions in general texts. A mention usually consists of two main parts: a position part and a content part. The later part may be a noun or a noun phrase, which determines the referred object. The former part conforms to some regular expressions which locate the position of the referred object. Referents (Antecedents) are definitions or explanations of related terms or provisions. They can be nouns, noun phrases, sentences, paragraphs of articles or even whole articles in some cases. They help readers fully comprehend the law, and also help lawmakers create concise and easy-to-understand legal texts.

Logical Parts and Logical Structures of Legal Texts

At the sentence level, a law sentence can roughly be divided into two *high-level*[4] logical parts: *requisite part* and *effectuation part* [5, 6, 110] in the form of:

$[requisite\ part] \Rightarrow\ [effectuation\ part]$

Each *requisite part* or *effectuation part* consists of several logical parts. A logical part is a clause or a phrase in a law sentence that conveys a part of the meaning of legal texts. Each logical part contains a specific kind of information according to its type. Three main types of logical parts are *antecedent part*, *consequent part*, and *topic part*. A logical part in consequent type describes a law provision; a logical part in antecedent type indicates cases (or the context) the law provision can be applied; and a logical part in topic type describes subjects related to the law provision. In a simple case[5], the *requisite part* only

[4]The reason why they call them high-level is that each *requisite part* or *effectuation part* consists of several logical parts.

[5]To understand more about four cases of legal sentences and their logical parts, please check the paper of Bach et al. [7]

Figure 5.2: An example of law sentences and their logical parts (A: Antecedent part; C: Consequent part; T: Topic part).

consists of a *topic part* or an *antecedent part*; and the *effectuation part* only consists of one *consequent part*.

Figure 5.2 shows four cases of law sentences and their logical parts. Logical structures in four cases can be expressed as follows:

At the paragraph level, a paragraph usually contains a main sentence and one or more subordinate sentences [109]. To be concrete, in a paragraph, the first sentence presents a law provision, and the other sentences describe cases in which the law provision can be applied.

5.3.2 A type of questions raised from the characteristics of legal texts

This sub-section describes a type of legal questions, which is mostly raised from above characteristics.

Generally speaking, to find correct answers, a QA system should have the ability to interpret content of documents. At the discourse level, legal documents are highly

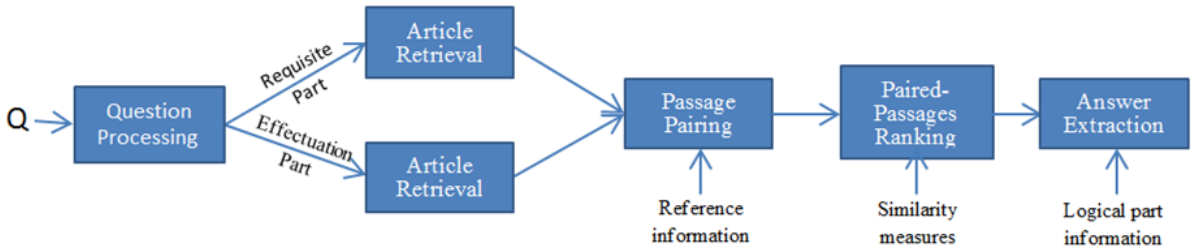

Figure 5.3: A framework to extract answers for a type of legal questions.

related by references, which usually bring precious information. A law cannot be correctly interpreted without reading some of the referenced items within it. These references can be placed on requisite parts or effectuation parts of sentences. This means that if a sentence contains a reference, the real content of its requisite or effectuation parts actually lie in a different document. For example, the sentence in '*article 12, paragraph 4*' of the tracing example has its requisite part lying in '*article 12, paragraph 1*' because it imports the definition of the notification in that document. The challenge for us is to be able to identify references and to jointly interpret them. Therefore, a good QA system should have the ability to follow these connections, which are represented via the relation of the references and their referents.

There are many legal questions falling under this type of questions because many users tend to ask about the beneficial conditions of laws, or the beneficiaries that can be achieved if some conditions of laws are satisfied [6].

5.4 A Proposed Framework for a Legal Question Answering System

Based on the above analyses, we propose a framework to help extracting answers to this type of legal questions as presented in Figure 5.3. This framework includes five steps. Each input question will be processed through the question processing step. In this step, each question is split into two parts, i.e. the requisite part and the effectuation part. In the next step - Article Retrieval, two collections of relevant articles are retrieved by using the content words and their synonyms of two parts respectively. Next, in the passage pairing step, a passage of articles in the first collection is aligned to a passage of articles in the second one by using the reference-referent if available. The result is a set of paired-passages which are likely to contain evidence for finding the correct answer. To find the best pair, we rank all pairs by using some effective similarity measures derived from previous research [63]. The best pair passage will be used to extract the correct answer to the input question by using logical structures of legal texts. In the following sub-sections, we will present the detail of each step in this framework.

Figure 5.4 illustrates a running example of the proposed system. The question was first processed via the first step to divide it into two parts, i.e. requisite part and effectuation part. The next step retrieved 2 article collections corresponding to two logical parts. With N equals to 5, C_r includes 5 articles (A12, A5, A105, A94-3, A10) and C_e includes

[6] We can use these characteristics for a QA system as shown in Tomura, K., A study on a question answering system for laws, Master thesis, JAIST, 2013. The system answers to a question based on only one document.

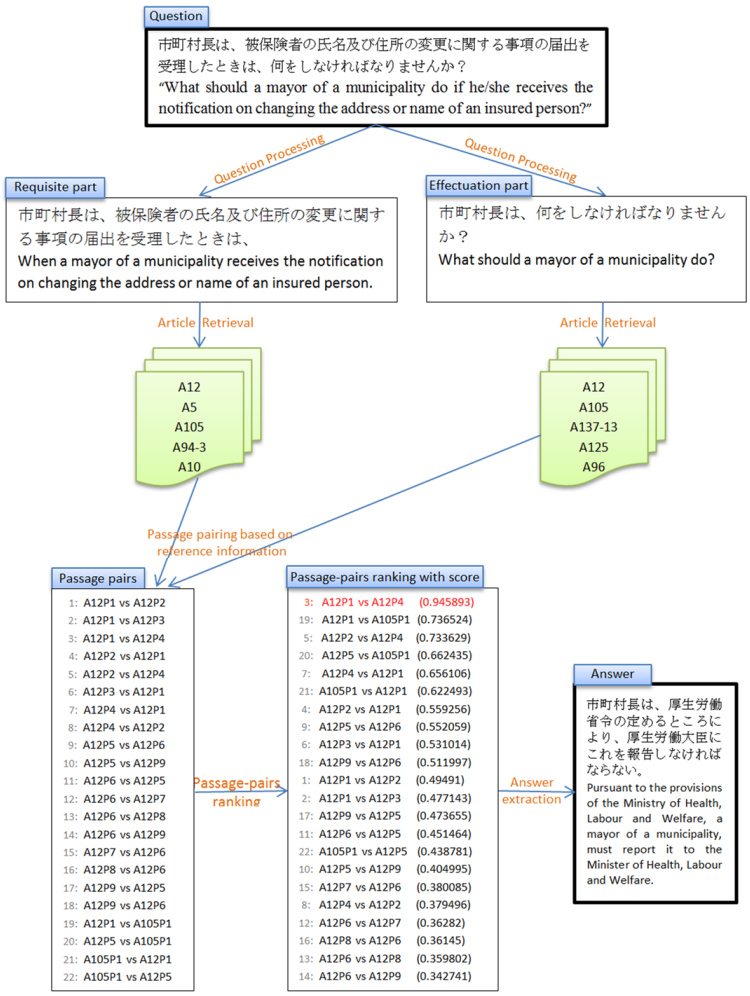

Figure 5.4: A true example of the proposed system.

5 articles (A12, A105, A137-13, A125, A96). Passages in the articles of C_r are paired with passages in the articles of C_e if they contain at least one reference which refers to the other passages or vice verse. This step leaded to the results including 22 passage pairs. The next step measured the similarity scores between these pairs and the question. The pair, A12P1 and A12P4, with the highest score is chosen as containing the answer. The question word lies in the effectuation part and the reference lies in A12P4, so the system extracted the effectuation part of A12P4 as the correct answer. In the following sub-sections, we will present in more details about these steps.

5.4.1 Question Processing

The goal of this part is to split the question into two parts, i.e. the requisite part q_r and the effectuation part q_e. We exploit an implementation of Bach et al. [7] to recognize these parts. An example is given in Figure 5.5. Each part is then preprocessed by word segmentation, POS tagging, and dependency parsing using Cabocha tool[7]. We keep content words and remove stop-words by using a list downloaded from this website[8].

[7]http://code.google.com/p/cabocha/

[8]http://www.ranks.nl/stopwords/japanese.html

question q	市町村長は、被保険者の氏名及び住所の変更に関する事項を受理したときは、何をしなければなりませんか？ *What should a mayor of a municipality do if he/she receives the notification on changing the address or name of an insured person?*
q after analyzing logical parts	<T3>市町村長は、</T3><A>被保険者の氏名及び住所の変更に関する事項を受理したときは、</A><C>何をしなければなりますか？</C>
Requisite part q_r	*(T3, A)* = when a mayor of a municipality receives the notification on changing the address or name of an insured person.
Effectuation part q_e	*(T3, C)* = What should a mayor of a municipality do?

Figure 5.5: An example of the question processing step (A: Antecedent part; C: consequent part; T: Topic part).

Removing stop-words helps the model ignore function words and high-frequency, but low-content words.

In fact, the forms and words in user's questions might be different from real laws 's. Therefore, the exact wording of the answers might look nothing like the questions. Thus, it is necessary to expand the question by adding terms in hopes of matching the particular from of the answer as it appears. In other words, to increase the number of relevant articles, we also use the synonyms of each keyword in the question by using a Japanese synonym list[9].

5.4.2 Article Retrieval

Based on the content words extracted from the previous step, we retrieved relevant articles from the corpus using Boolean *AND* and *OR* queries. The information retrieval system selects a set of potentially relevant articles that are likely to contain the evidence for finding correct answers. To retrieve, we implemented cosine similarity between the question and an article. In the vector space model [47], articles and questions are represented as vectors of features representing the terms (keywords) that occur within the collection. The value of each feature is called the *term weight*. In this system, we use conventional tf-idf [65] term weighting which is very useful and popular in many information retrieval tasks [81, 83, 127]. Particularly, the weight of term i in the vector for article d is:

$$w_{i,j} = tf_{i,j} \times idf_i \tag{5.1}$$

where $tf_{i,j}$ is term frequency of term i in article j; idf_i is inverse document frequency and calculated using the following equation:

$$idf_i = log\frac{N}{n_i} \tag{5.2}$$

where N is the total number of articles in the collection, and n_i is the number of articles in which term i occurs.

Therefore, the cosine similarity between the question $\vec{q}$ and the article $\vec{d}$ is calculated using the following equation:

[9]We used synonym list extracted from Japanese WordNet Copyright 2009, 2012 by National Institute of Information and Communications Technology (NiCT).

$$sim(\vec{q},\vec{d}) = \frac{\sum_{w\in q,d} tf_{w,q}tf_{w,d}(idf_w)^2}{\sqrt{\sum_{q_i\in q}(tf_{q_i,q}idf_{q_i})^2} \times \sqrt{\sum_{d_i\in d}(tf_{d_i,d}idf_{d_i})^2}} \tag{5.3}$$

This step returns two article sets C_r and C_e. C_r contains a set of relevant articles for the requisite part of the question, q_r. C_e contains a set of relevant articles for the effectuation part of the question, q_e.

5.4.3 Passage Pairing

The purpose of this step is to link passages of articles in two sets C_r and C_e using the reference-referent information. Two passages in two sets are linked if one passage contains at least one reference, which refers to a referent in the other passage. In more detail, we pair each paragraph p_r in an article of the set C_r to a paragraph p_e of an article in the set C_e if there exists one reference in p_r referring to a fragment of texts in the paragraph p_e and vice verse.

5.4.4 Paired-Passages Ranking

In this step, all pairs in the form of (p_r, p_e) are ranked using a ranking function. The ranking function is a linear combination of some similarity scores between the passage pair and the question. The similarity score of each passage pair (p_r, p_e) with the question (q_r, q_e) is calculated using the following equation:

$$TotalScore((p_r,p_e),(q_r,q_e)) = TotalScore(p_r,q_r) + TotalScore(p_e,q_e) \tag{5.4}$$

Each $TotalScore(,)$ between an answer passage p_x and a question part q_x is calculated using the following equation:

$$TotalScore(p_x,q_x) = \sum_{i=1}^{n} \lambda_i \times score_i(p_x,q_x) \tag{5.5}$$

where λ_i is the weight of $score_i$; each $score_i(p_x, q_x)$ corresponds to one score in the following sets of scores derived from the work of Surdeanu et al.[63]. For the sake of simplicity, we set all λ_i equal to 1.

- *Similarities*

  The similarity between an part of a question q and the passage p is measured using the length-normalized *BM25* formula [100, 101]. According to this score, the similarity between q and p is calculated as follows:

  $$BM25(p) = \sum_{i=0}^{|q|} \frac{(k_1+1)tf_{i,p}(k_3+1)tf_{i,q}}{(K+tf_{i,p})(k_3+tf_{i,q})} log(idf_i) \tag{5.6}$$

  where $tf_{i,p}$ and $tf_{i,q}$ are the term frequencies of the question term i in q and p; and idf_i is the inverse document frequency of term i in the answer passage collection. K is the length-normalization factor:

$$K = k_1((1-b) + b\,|A|\,/avg\_len)$$

where *avg_len* is the average answer passage length in the collection. For all the constants in the formula we also use values reported optimal for other IR collections [63] ($b = 0.75, k_1 = 1.2$, and $k_3 = 1,000$).

For completeness, we also include the value of the $tf.idf$ similarity measure as presented in the article retrieval step.

To understand the contribution of the syntactic and semantic processors, we compute the above similarity measures using three different representations of the question and passage content as follows:

- *Words (W)* - the text is considered as a bag of words.
- *Dependencies (D)* - the text is represented as a bag of binary syntactic dependencies. We extract dependency paths of length 1, i.e., direct head-modifier relations.
- *Bigrams (B)* - the text is represented as a bag of bi-grams. This view is added to ensuring a fair analysis of the above syntactic views.

- *Density and frequency scores*

  These scores measure the density and frequency of question terms in the passage text.

  - *Same word sequence* - computes the number of non-stop question words that are recognized in the same order in the passage.
  - *Answer span* - the largest distance (in words) between two non-stop question words in the passage.
  - *Same sentence match* - number of non-stop question terms matched in a single sentence in the passage.
  - *Overall match* - number of non-stop question terms matched in the complete passage. These scores are normalized into [0,1]. These last two scores are computed also for the other two text representations previously introduced above.
  - *Informativeness* - models the amount of information contained in the answer passage by counting the number of non-stop nouns, verbs, and adjectives in the passage that do not appear in the question.

5.4.5 Answer Extraction

At the paragraph level, a paragraph usually contains a main sentence and one or more subordinate sentences. In this thesis, we used an implementation of Bach et al. [5, 7] to recognize the logical structures of paragraphs to extract the answer.

<T3>市町村長は、</T3><A>第一項又は第二項の規定による届出を受理したときは、</A><C>厚生労働省令の定めるところにより、厚生労働大臣にこれを報告しなければならない。</C>
<T3> a mayor of a municipality, </T3> <A>when receiving *the notification in the provision of Para 1 or Para 2* </A>, <C>pursuant to the provisions of the Ministry of Health, Labour and Welfare, s/he must report it to the Minister of Health, Labour and Welfare. <C>

Figure 5.6: An example of the answer extraction step (A: Antecedent part; C: consequent part; T: Topic part).

- If the question part lies in the effectuation part of the question, we extract the effectuation part of the main paragraph[10] as the answer and vice versa. To determine this, we use clues of question words such as *dare, itsu, doko, desuka, masenka, masuka, dono, nani, etc.*
- If the answer contains references, we also extract their referents to help people fully understanding it.

Figure 5.6 shows an example of an answer sentence after analyzing its logical structure. Because the question part asks about the consequence of an action, we extract its consequence part (consisting of the topic part and the consequent part) as the final answer *'pursuant to the provisions of the Ministry of Health, Labor and Welfare, s/he must report it to the Minister of Health, Labor and Welfare'*

5.5 Experimental Results of the QA system

In this section, we first present experimental setups including testing data, evaluation measure and a traditional QA system. The purpose of implementing this traditional QA system is to compare the performance of our QA system using reference information and not using them. Then, we present experimental results of our QA system using the proposed method and a traditional QA system.

5.5.1 Experimental Setups

This sub-section presents the testing legal questions and evaluation measure to estimate the system's performance. We also briefly present a traditional QA system to prove that using reference information in answering this special type of questions yields better results.

Data

We tested our system using 51 legal questions on the Japanese National Pension Law. To help us understand more about the behavior of the systems, we categorize these questions into two main classes based on how they use the reference information in determining the answers.

1. Obligatory references-resolving questions: Relevant sentences, which provide evidence to answers, lie in different documents. These sentences are linked through the reference information. This class is sub-divided as follows:

[10] the paragraph which contains the reference referring to the referenced paragraph.

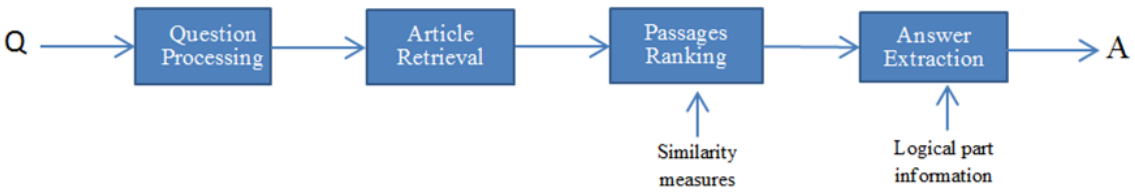

Figure 5.7: The framework of the traditional QA system.

(a) Bi-document-linking questions: Only two documents are linked using reference information to provide evidence to answers. A majority of legal questions falls into this case.

(b) Multi-document-linking questions: more than two documents are linked to find answers using reference information.

2. Optional references-resolving questions: In this case, the referenced documents play the role of explaining more about the terms/phrases in the users questions. Therefore, QA systems can still find the answers without using reference information.

The class of each question is given in Table 5.1.

Evaluation measure

To evaluate the performance, we use the evaluation measure of ResPubliQA 2009 [86] which is a QA evaluation task over European Legislation, proposed at the Cross Language Evaluation Forum (CLEF 2009). Because the traditional system and the proposed system output answers to all 51 questions, the evaluation measure $c@1$ [86] becomes the accuracy measure calculated as follows:

$$Accuracy = \frac{\#CorrectlyAnsweredQuestions}{\#Questions} \tag{5.7}$$

A traditional QA system

The traditional QA system consists of four steps as presented in Figure 5.7. In the question processing step, the question is processed as same as in the proposed method except for dividing it into its logical parts. The second step, article retrieval, retrieves top N relevant articles as in our method by using all question words and their synonyms. The third step, the passages ranking, ranks all passages in each relevant articles based on their similarity scores with the question. In this step, we use all two sets of scores as in our methods. The last step is as same as the answer extraction step of the proposed method.

5.5.2 Experimental results using the traditional QA system and the proposed system

Table 5.1 presents the experimental results using the traditional QA system and the proposed system on 51 legal questions. This table also shows the class of each question based on the above classification. Table 5.2 presents the experimental results of the

traditional QA and the proposed QA systems using two evaluation criteria. The first criterion is to measure the performance based on the extracted paragraphs. This means that if the systems correctly determine the main paragraphs (which contain the answers). The second criterion is to measure the performance based on the extracted answers. This means that if the systems correctly find the answers.

Next, we describe the performance of each QA system in more details.

The traditional QA system

Table 5.1: Experimental results of two QA systems using the traditional method and the proposed method on 51 legal questions.

Question No.	Traditional Method	Proposed Method	Questions' Class
0	Wrong (referenced paragraph)	True	1a
1	True	True	2
2	Wrong (referenced paragraph)	Wrong	1a
3	Wrong (referenced paragraph)	True	1a
4	Wrong (referenced paragraph)	True	1a
5	Wrong (referenced paragraph)	Wrong	1b
6	True	True	1a
7	Wrong (referenced paragraph)	True	1a
8	Wrong	True	1a
9	Wrong (referenced paragraph)	True	1b
10	Wrong	Wrong	1b
11	Wrong	Wrong	1a
12	Wrong	Wrong	1a
13	True	True	1b
14	Wrong (referenced paragraph)	True	1b
15	Wrong (referenced paragraph)	True	1a
16	Wrong	Wrong	1b
17	Wrong (referenced paragraph)	True	1a
18	Wrong	Wrong	1a
19	Wrong (referenced paragraph)	Wrong	1a
20	Wrong (True main paragraph)	True	1a
21	True	True	1a
22	Wrong (referenced paragraph)	Wrong	1a
23	Wrong	Wrong	1a
24	Wrong (referenced paragraph)	Wrong	1a
25	True	True	1a
26	Wrong (True main paragraph)	True	1a
27	Wrong (referenced paragraph)	True	1a
28	True	True	1a
29	Wrong (referenced paragraph)	Wrong	1a
30	Wrong	True	1a
31	True	Wrong	1b
32	True	True	1a
33	Wrong (referenced paragraph)	True	1a
34	Wrong (referenced paragraph)	True	1a
35	Wrong (referenced paragraph)	Wrong	1b
36	Wrong (referenced paragraph)	True	1a
37	Wrong (referenced paragraph)	True	1b
38	Wrong (referenced paragraph)	Wrong	1b
39	True	True	1a
40	Wrong	Wrong	1b
41	Wrong	Wrong	1b
42	True	True	1a
43	Wrong (referenced paragraph)	Wrong	1a
44	Wrong (referenced paragraph)	True	1a
45	True	Wrong (True main paragraph)	1a
46	Wrong	True	1a
47	Wrong (referenced paragraph)	Wrong	1a
48	Wrong	True	1a
49	True	True	1a
50	Wrong (True main paragraph)	True	1b

Table 5.2: Accuracy of the QA system using two methods on 51 questions.

	Traditional System		**Proposed System**	
	Paragraphs	**Answers**	**Paragraphs**	**Answers**
#CorrectQuestions	15	12	**32**	**31**
Accuracy(%)	29.4	23.5	**62.7**	**60.8**

Although, the traditional QA system did not use the reference information, it still correctly found the main paragraphs for 15 questions and correctly extracted answers for 12 questions. The reasons are high word overlaps between the questions and the main paragraphs. Another reason is that in some questions, the usage of reference information is optional to the process of finding their answers (i.e. in Question 2 in Figure 5.8, it is not necessary to link the information from the main para A92-4P2 to the referenced para A92-4P1. But, the information in A92-4P1 helps us understand more about the situation). There are 24 questions that the traditional system finds out the referenced paragraphs instead of the main paragraphs (i.e. the tracing example question). The reason is that their majorities of question words contained in the referenced paragraphs (as in the example question). Because their answers are not contained in these paragraphs, the system is impossible to extract their correct answer.

For the remaining questions, the traditional system could not determine relevant paragraphs. Hence, it is unable to find their answers.

In questions 20, 26, and 50, the system correctly finds the main paragraphs. However, because the correct answers lie in the referents of the references contained in these main paragraphs, it cannot extract their correct answers. The accuracy of the traditional system, therefore, is 23.5%.

The proposed QA system

There are 20 questions whose answers could not be found. The reason may be that the similarity measures could not capture the entire context between questions and the paragraphs, which contain the answers; or the errors of the processing tools. For examples, in question 48, the system correctly determines the paragraph pair, which contains the answer. However, it extracts the wrong answer because of the error of the requisite-effectuation tool[11] (in Q.48, the answer is '*14.6% per year*' instead of the extracted answer bounded in tags ⟨C⟩ and ⟨/C⟩.

There are 13 questions, in which finding their answers requires that the main paragraph must be linked to more than one document to provide the contexts for the correct answers (i.e. to find the answer of Question 5 in Figure 5.4, it is necessary to link the information from the document A96P3 to the document A96P1 via the document A96P2). Although the proposed framework does not allow us to process on more than two documents, it can still find the correct main paragraph containing the correct answers (in 5 questions). In these cases, the system correctly determines one linking pair between the main paragraph and one of the referenced paragraphs. Because the main paragraphs are correctly determined, the proposed method can extract the correct answers. This method also provides concrete evidences of the answers by showing the paragraph pairs, which contain the answers.

[11]This tool got the accuracy of $\sim$ 90%

Q. 48	The answer in 第 97 条第 1 項 Article 97, Paragraph 1 <A>前条第一項の規定によつて督促をしたときは、</A><T3>厚生労働大臣は、</T3><A>徴収金額に、納期限の翌日から徴収金完納又は財産差押の日の前日までの期間の日数に応じ、年十四・六パーセント（当該督促が保険料に係るものであるときは、</A><T2>当該納期限の翌日から三月を経過する日までの期間については、</T2><C>年七・三パーセント）の割合を乗じて計算した延滞金を徴収する。</C>ただし、<A>徴収金額が五百円未満であるとき、又は滞納につきやむを得ない事情があると認められるときは、この限りでない。</A> When the payment is demanded pursuant to the provision of the preceding paragraph of the preceding article, Minister of Health, Labour and Welfare collects money in arrears calculated by multiplying the amount of money to be collected by the ratio of 14.6 percent a year (when the demand for payment concerns insurance premium, it is 7.3 percent a year for terms by the day when three months passes from the next day of the deadline), depending on the number of days of terms from the next day of the deadline for payment until the previous day of the day of full payment or attachment of property; provided, however, that this shall not apply when the amount of money to be collected is less than 500 yen, or when unavoidable circumstances with respect to delinquency are found.
Q. 5	厚生労働大臣は、保険料その他この法律の規定による徴収金を滞納する者に対して督促することができますが、納付義務者に対して、督促できるタイミングはいつですか？ Though the Minister of Health, Labour and Welfare may demands a person who fails to pay insurance premium or other money to be collected pursuant to the provisions of this law, when is it that s/he may demand the person who owes to pay?
Clues	第 96 条第 3 項 - Article 96 Paragraph 3 前項の督促状により指定する期限は、督促状を発する日から起算して十日以上を経過した日でなければならない。 The deadline designated by the demand letter mentioned in the previous paragraph must be the day when more than ten days passed since the day when the demand letter was issued. 第 96 条第 2 項 - Article 96 Paragraph 2 前項の規定によつて督促をしようとするときは、厚生労働大臣は、納付義務者に対して、督促状を発する。 When the Minister of Health, Labour and Welfare demands pursuant to the provision of the previous paragraph, s/he issues a demand letter to the person who owes to pay. 第 96 条第 1 項 - Article 96 Paragraph 1 保険料その他この法律の規定による徴収金を滞納する者があるときは、厚生労働大臣は、期限を指定して、これを督促することができる。 When a person fails to pay insurance premium or other money to be collected pursuant to the provisions of this law, the Minister of Health, Labour and Welfare may demand it designating the deadline.
Q. 2	納付受託者は、被保険者から保険料の交付を受けたときは、何をしなければなりませんか？ What does the payment trustee have to do, when they received delivery of premium from the insured?
Clues	第 92 条の 4 第 2 項 - Article 92-4 Paragraph 2 納付受託者は、前項の規定により被保険者から保険料の交付を受けたときは、遅滞なく、厚生労働省令で定めるところにより、その旨及び交付を受けた年月日を厚生労働大臣に報告しなければならない。 When the payment trustee received the delivery of the insured premium from the insured person pursuant to the provisions of the preceding paragraph, without delay, pursuant to the provisions of the Ministry of Health, Labour and Welfare, it must report to the Minister of Health, Labour and Welfare that the insured premium was delivered and the date of the delivery. 第 92 条の 4 第 1 項 - Article 92-4 Paragraph 1 被保険者が前条第一項の委託に基づき保険料を同項各号に掲げる者で納付事務を行うもの（以下「納付受託者」という。）に交付したときは、納付受託者は、政府に対して当該保険料の納付の責めに任ずるものとする When the insured person delivered based on the entrustment mentioned in the first paragraph of the previous article the insured premium to one of those (from now on called the payment trustee) who are listed in each item of the same paragraph and who perform payment affairs, the payment Trustees shall be responsible for the payment of such insured premium to the government.

Figure 5.8: Some typical examples of the systems.

The accuracy of the proposed system, therefore, is 60.8%. It can be seen that the proposed system outperformed the traditional system, which did not exploit the reference information. Even if the traditional system can find the main paragraph because of high word overlaps, it cannot provide the evidence to help users believe in the systems output. However, our method can do this.

5.6 Conclusion and Future Work

This chapter presented an application of reference information to build a legal QA system. We focused on one type of questions whose answers can not be found from merely one document. To find their correct answers, it is necessary to link documents via the relation of reference-referent. To achieve the goal, we first built a reference resolver. Based on that, we proposed a novel framework which allows us to exploit the reference information between legal documents to find answers. This approach also uses the requisite-effectuation structures of legal sentences and some effective similarity measures based on legal terms to

support finding correct answers without training data. The experimental results showed that the proposed method was quite promising and outperformed a traditional method which did not use reference information.

In our framework, there is an assumption that questions and related paragraphs can be divided into two parts. Therefore, the proposed system is restricted to legal questions asking about the requisite and the effectuation problems. In fact, there are many questions falling under this category because users tend to ask about the beneficial conditions of laws, or the beneficiaries that can be achieved if some conditions are satisfied. As an initial step, we selected these questions manually. In the future, we aim at building a question classifier, which can automatically filter this type of questions. In addition, the assumption about dividing a paragraph into its logical structure is also quite reasonable. We counted the frequency of paragraphs having requisite-effectuation structures in the JPL corpus which are not definitions, and got 537 paragraphs among 547 paragraphs[12]. Hence, the ratio of paragraphs having the requisite-effectuation structure is very high (98.2%). In our corpus, definition sentences are also marked using requisite and effectuation tags where a defined term is an effectuation and an explanation part a requisite. Therefore, our method is also applied to definition paragraphs. Another aspect is that we focused on providing the QA system with questions which are more easier to find the answers. In fact, natural questions are usually ambiguous, therefore they need complicated preprocessing techniques. These two problems will be further considered in the future work. Moreover, we also plan to extend the framework so that the system can handle more than two-linked documents.

[12]We did not count the number of definition in parentheses and only count paragraph main sentences.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we investigated the task of reference resolution and its application to legal question answering. Reference resolution is a task of determining which entities are referred to by which linguistic expressions. This phenomenon is not only popular in general texts but also in legal texts. Once this task is solved, it does not only help readers in comprehending texts well, but also in supporting other tasks in Natural Language Processing such as machine translation, text summarization, information retrieval and question answering. Among six chapters, the main chapters are Chapters 3, 4, and 5. The main contributions of our work can be summarized as follows:

- Firstly, we investigated the task of coreference resolution in general texts. This problem has received much attention of researchers. Previous work has a drawback which only considers one or two candidates . Therefore, the probability assigned to each candidate merely encodes the likelihood of that particular candidate being coreferential with a given mention. To overcome this drawback, we proposed a listwise approach using learning-to-rank algorithms. This listwise approach allows all candidates to be examined simultaneously. Experimental results on a shared task corpus showed the effectiveness of the proposed approach. In comparison to the best participating system SUCRE, which uses the Decision Tree algorithm with the best-first clustering strategy, our proposed system achieved comparable performance. These results demonstrated that the listwise approach is appropriate for the coreference resolution task.

- Secondly, we investigated the task of reference resolution in the legal domain. The goal was to create a system which can automatically detect references and then extracts their referents. This is a new interesting task in Legal Engineering research. Previous work limited itself to detect and resolve references at the document targets. We go a step further by trying to resolve references to sub-document targets. Referents extracted are the smallest fragments of texts in documents, rather than the entire documents that contain the referenced texts. Based on analyzing the characteristics of reference phenomena in legal texts, we propose a four-step framework to accomplish the task. We also show how machine learning methods can be exploited in each step. The final system achieves 80.06% in the F1 score for detect-

ing references, 85.61% accuracy for resolving them, and 67.02% in the F1 score on the end-to-end setting task on the Japanese National Pension Law corpus.

- Finally, we presented a study on exploiting reference information to build a question answering system restricted to the legal domain. Most previous research focuses on answering legal questions whose answers can be found in one document without using reference information. However, there exist many legal questions, which require answers extracted from connections of more than one document. To the best of our knowledge, this type of questions is not adequately considered in previous research. To cope with them, we propose a novel approach which allows exploiting the reference information between legal documents to find answers to this type of legal questions. This approach also uses the requisite-effectuation structures of legal sentences and some effective similarity measures based on legal terms to support finding correct answers without training data. The experimental results showed that the proposed method is quite effective and outperformed a baseline method which does not use reference information.

The contribution of this dissertation also includes linguistic and computational aspects. From the linguistic viewpoint, our research helps in interpreting the sentences of any discourse. From the computational viewpoint, our research proposes effective solutions for linguistic problems using machine learning approaches.

6.2 Future Work

In future work, we plan to focus on remaining issues of this thesis. The research in this dissertation can be extended in many directions.

Firstly, we aim to continue improving the performance of the reference resolvers as well as the question answering system. To perform this, we will complete our framework to make it stronger. To accomplish these goals, we intend to pursue the following research directions:

- Two listwise learning-to-rank algorithms are considered in Chapter 3. There are still some other algorithms such as AdaRank [123], which have not been investigated in this thesis. In the future, we will endeavor to implement other algorithms to complete the framework. We also shall conduct experiments of the proposed approach on other public corpora to estimate its performance.

- Concerning the task of reference resolution in legal texts, there a wide range of open possibilities to improve the system. Our models use machine-learning approaches, therefore, features play an important role. In the future, we will integrate more features to improve the system, such as features extracted from outside resources, etc. We will also consider problems such as versioning, etc.

- To complete the QA system in the legal domain, we will collect more questions to test the performance of the system. We will also extend our work in order to be able to handle more than two linked documents.

Secondly, we aim at extending our work to other types of legal texts, rather than the Japanese National Pension Law. To adapt our system to work on other laws, it is necessary to investigate those laws to understand the naming rules of the law systems. In other words, our system should change towards understanding the structures of laws. For other parts of the frameworks, we think that our approach is able to work well on other types of legal texts. Moreover, once we have been successful in developing corresponding systems for Japanese, we could think to extend our system to multi-language systems, which can operate in other languages such as Vietnamese, English, etc. Our purpose is to build real systems that can support users in easily accessing and fully understanding as many kinds of natural texts as possible.

Finally, we also aim at investigating the more effective effects of reference information on other applications of natural language processing, such as text summarization, and finding contradictions in legal texts.

Appendix A

Questions and Answers List

No.	Questions	Gold Answers	Our System's Answers
0	市町村長は、被保険者の氏名及び住所の変更に関する事項の届出を受理したときは、何をしなければなりませんか？	厚生労働省令の定めるところにより、厚生労働大臣にこれを報告しなければならない。	厚生労働省令の定めるところにより、厚生労働大臣にこれを報告しなければならない。
1	障害基礎年金の額が改定されたときは、改定後の額による障害基礎年金の支給は、いつから始めるものとされていますか？	改定が行われた日の属する月の翌月から始めるものとされている。	改定が行われた日の属する月の翌月から始めるものとする。
2	被保険者は、将来の一定期間の保険料を前納することができますが、その場合に前納できる額は、どのようになりますか？	当該期間の各月の保険料の額から政令で定める額を控除した額。	Wrong
3	各年金保険者たる共済組合等は、誰を経由して、当該年金保険者たる共済組合等に係る被保険者の数その他の厚生労働省令で定める事項を厚生労働大臣に報告するのですか？	当該年金保険者たる共済組合等を所管する大臣。	厚生労働省令の定めるところにより、当該年金保険者たる共済組合等を所管する大臣を経由して前項の報告を行うものとする。 And its reference to A94-5P1 厚生労働大臣は、年金保険者たる共済組合等に対し、当該年金保険者たる共済組合等を所管する大臣を経由して、当該年金保険者たる共済組合等に係る被保険者の数その他の厚生労働省令で定める事項について報告を求めることができる。
4	保険料その他この法律の規定による徴収金を滞納する者があるときは、厚生労働大臣は、期限を指定して、これを督促することができますが、督促をしようとするときは、厚生労働大臣は、何をする必要がありますか？	納付義務者に対して、督促状を発する。	納付義務者に対して、督促状を発する
5	厚生労働大臣は、保険料その他この法律の規定による徴収金を滞納する者に対して督促することができますが、納付義務者に対して、督促できるタイミングはいつですか？	督促状を発する日から起算して十日以上を経過した日から。	Wrong
6	被保険者の資格に関する処分の取消しの訴えは、何を経なければ提起できませんか？	当該処分についての再審査請求に対する社会保険審査会の裁決。	当該処分についての再審査請求に対する社会保険審査会の裁決を経た後でなければ、提起することができない。

Figure A.1: This is a list of questions with their gold answers and the proposed system's answers.

7	年金給付を受ける権利は、その支給事由が生じた日から五年を経過したときは、時効によって消滅しますが、時効が進行しないのは、どんなとき？	当該年金給付がその全額につき支給を停止されているとき。	当該年金給付がその全額につき支給を停止されている間は、
8	地域型基金の設立委員は、規約を作成し、創立総会の日時及び場所とともに公告して、創立総会を開かなければなりませんが、その公告は、いつまでにしなければなりませんか？	会日の二週間前までにしなければならない。	会日の二週間前までにしなければならない。
9	国民年金基金の代議員会の理事長は、代議員会が成立しないとき、又は理事長において代議員会を招集する暇がないと認めるときは、代議員会の議決を経なければならない事項で臨時急施を要するものを処分することができますが、その処置実施後、理事長は何をしなければなりませんか？	次の代議員会において実施した処分を報告し、その承認を求めなければなりません。	前項の規定による処置については、次の代議員会においてこれを報告し、その承認を求めなければならない。 And its reference to A123P2 理事長は、代議員会が成立しないとき、又は理事長において代議員会を招集する暇がないと認めるときは、代議員会の議決を経なければならない事項で臨時急施を要するものを処分することができる。
10	国民年金基金連合会を設立するにあたって、創立総会の公告は、いつまでにしなければなりませんか？	会日の二週間前までにしなければならない。	Wrong
11	遺族基礎年金の受給権者が直系血族又は直系姻族以外の養子となつたときは遺族基礎年金の受給権はどうなりますか？	消滅する。	Wrong
12	受給権者が支給停止の申出を撤回したい場合、いつまでならできますか？	いつでもできます。	Wrong
13	政府は、教育及び広報を行うことを誰に行わせることができるでしょうか？	日本年金機構に行わせることができる。	第一項各号に掲げる事業及び前項に規定する運用の全部又は一部を日本年金機構（以下「機構」という。）に行わせることができる。 And its reference to A74P1 政府は、国民年金事業の円滑な実施を図るため、国民年金に関し、次に掲げる事業を行うことができる。。。
14	厚生労働大臣は、年金積立金管理運用独立行政法人に対し積立金を寄託するまでの間、どこに積立金を預託することができるのですか？	財政融資資金に積立金を預託することができる。	前項の規定にかかわらず、同項の規定に基づく寄託をするまでの間、財政融資資金に積立金を預託することができる And its references to A76P1 積立金の運用は、厚生労働大臣が、前条の目的に沿つた運用に基づく納付金の納付を目的として、年金積立金管理運用独立行政法人に対し、積立金を寄託することにより行うものとする。
15	厚生労働大臣は、被保険者から指定代理納付者をして当該被保険者の保険料を立て替えて納付させることを希望する旨の申出を受けたときは、どんなときに限り、その申出を承認することができるのですか？	指定代理納付者による納付が確実と認められ、かつ、指定代理納付者による納付希望の申出を承認することが保険料の徴収上有利と認められるときに限りその申し出を承認することができる。	その納付が確実と認められ、かつ、その申出を承認することが保険料の徴収上有利と認められるときに限り、その申出を承認することができる
16	国民年金法第七条第一項各号において、国内居住要件が規定されている被保険者は第何号被保険者？	第一号被保険者。	Wrong

Figure A.1 (continued)

17	障害若しくは死亡又はこれらの直接の原因となつた事故が第三者の行為によつて生じた場合において、受給権者が第三者から同一の事由について損害賠償を受けたときは、政府の給付責任はどうなりますか？	損害賠償の価額の限度で、給付を行う責を免かれる。	その価額の限度で、給付を行う責を免かれる。
18	法律上の婚姻関係はないが、事実上婚姻関係と同様の事情にある者の一方は、被保険者たる他方の保険料に対してどのような義務を負いますか？	連帯して納付する義務を負います。	Wrong
19	年金給付の受給権者が死亡した場合において、その死亡した者に支給すべき年金給付でまだその者に支給しなかつた年金があるとき、その未支給の年金を受けるべき者の順位はどのように定められていますか？	その死亡した者の配偶者、子、父 母、孫、祖父母、兄弟姉妹という順に定められています。	Wrong
20	障害の程度が障害等級の一級に該当する者に支給する障害基礎年金の額は、ある額の百分の百二十五に相当する額であるが、ある額とは？	七十八万九百円に改定率を乗じて得た額。	前項の規定にかかわらず、同項に定める額の百分の百二十五に相当する額とする And its reference to A33P1 障害基礎年金の額は、七十八万九百円に改定率を乗じて得た額（その額に五十円未満の端数が生じたときは、これを切り捨て、五十円以上百円未満の端数が生じたときは、これを百円に切り上げるものとする。）とする。
21	障害基礎年金の受給権者が、厚生労働大臣に対し、障害の程度が増進したことによる障害基礎年金の額の改定を請求する場合、障害基礎年金の受給権を取得した日又は厚生労働大臣の診査を受けた日から起算してどの程度経過した日後でないと請求出来ないか？	一年を経過した日後。	障害基礎年金の受給権を取得した日又は第一項の規定による厚生労働大臣の診査を受けた日から起算して一年を経過した日後でなければ
22	寡婦年金は、夫の死亡について労働基準法の規定による遺族補償が行われるべきものであるときは、死亡日からどの位の期間その支給を停止しますか？	六年間。	Wrong
23	運用職員がその職務に関して知り得た秘密を漏らし、又は盗用したと認めるときは、厚生労働大臣はどのような処置をしなければなりませんか？	その職員に対し国家公務員法に基づく懲戒処分をしなければならない。	Wrong
24	受給権者が、正当な理由がなくて、厚生労働省令の定める事項の届出をせず、又は厚生労働省令の定める書類その他の物件を提出しないときは、どのような処置をすることができますか？	年金給付の支払を一時差し止めることができる。	Wrong
25	納付事務を適正かつ確実に実施することができると認められ、かつ、政令で定める要件に該当する者として厚生労働大臣が指定するものは、その名称及び住所並びに事務所の所在地を変更しようとするときは、あらかじめ、その旨を誰に届け出なければならないでしょうか？	厚生労働大臣。	あらかじめ、その旨を厚生労働大臣に届け出なければならない。
26	年金保険者たる共済組合等は、何の算定のために、所管する大臣を経由して、厚生労働省令で定める事項を厚生労働大臣に報告するのでしょうか？	年金保険者たる共済組合等が納付すべき基礎年金拠出金について、その将来にわたる予想額。	厚生労働省令の定めるところにより、当該年金保険者たる共済組合等を所管する大臣を経由して、第九十四条の二第三項に規定する予想額の算定のために必要な事項として厚生労働省令で定める事項について厚生労働大臣に報告を行うものとする。 And its reference to A9-2P3 財政の現況及び見通しが作成されるときは、厚生労働大臣は、厚生年金保険の管掌者たる政府が負担し、又は年金保険者たる共済組合等が納付すべき基礎年金拠出金について、その将来にわたる予想額を算定するものとする。

Figure A.1 (continued)

27	被保険者の資格に関する処分に不服がある者が社会保険審査官に対してする審査請求は、時効の中断に関してはどのような扱いになるのでしょうか？	裁判上の請求とみなされます。	時効の中断に関しては、裁判上の請求とみなす。
28	国民年金事務組合は、当該構成員である被保険者の委託を受けようとするときは、厚生労働大臣の何を受けなければならないでしょうか？	認可。	厚生労働大臣の認可を受けなければならない。-
29	財務大臣は、厚生労働大臣から委任された納付義務者に係る滞納処分等その他の処分の権限を誰に委任するのですか？	国税庁長官。	Wrong
30	厚生労働大臣は、被保険者若しくは被保険者であつた者又は受給権者に係る保険料の納付に関する実態その他の厚生労働省令で定める事項に関して必要な調査を実施するに際して、必要があると認めるときは、どこに対し必要な情報の提供を求めることができるのですか？	官公署。	官公署に対し、必要な情報の提供を求めることができる。
31	保険料納付確認団体が被保険者の保険料滞納事実の有無について確認し、その結果を当該被保険者に通知する業務を適正に行うために、厚生労働大臣は保険料納付確認団体の求めに応じ、必要な限度で何ができますか？	保険料滞納事実に関する情報を提供することができる。	Wrong
32	解散した国民年金基金が、正当な理由がなくて、国民年金基金の解散に伴う責任準備金相当額の徴収金を督促状に指定する期限までに納付しないとき	六月以下の懲役又は五十万円以下の罰金に処せられる。	その代表者、代理人又は使用人その他の従業者でその違反行為をした者は、六月以下の懲役又は五十万円以下の罰金に処する。
33	地域型基金を設立するには、加入員たる資格を有する者及び年金に関する学識経験を有する者のうちから厚生労働大臣が任命した者が設立委員とならなければなりませんが、その設立委員の任命はどんな場合に行われるのですか？	三百人以上の加入員たる資格を有する者が厚生労働大臣に地域型基金の設立を希望する旨の申出を行つた場合。	三百人以上の加入員たる資格を有する者が厚生労働大臣に地域型基金の設立を希望する旨の申出を行つた場合に
34	職能型基金を組織する、第一号被保険者であつて、基金の地区内において同種の事業又は業務に従事する者を何といいますか？	加入員たる資格を有する者という。	加入員たる資格を有する者という。
35	基金の加入員たる資格を有する者であつて創立総会の会日までに設立委員等に対し設立の同意を申し出たものは、基金が成立したときは、どのタイミングで加入員の資格を取得しますか？	基金の成立の日。	Wrong
36	職能型基金の業務の種類を変更するには、どのような手続きを経なければその効力が生じないのでしょうか？	厚生労働大臣の認可手続き。	厚生労働大臣の認可を受けなければ、その効力を生じない。
37	理事が年金及び一時金に充てるべき積立金の管理及び運用に関する基金の業務を執行する場合において、その任務を怠つたときは、その理事はどうなりますか？	基金に対し連帯して損害賠償の責めに任ずることになる。	その理事は、基金に対し連帯して損害賠償の責めに任ずる。
38	基金は、理事が自己又は当該基金以外の第三者の利益を図る目的をもつて、積立金の管理及び運用の適正を害するものとして厚生労働省令で定める禁止行為をしてしまった場合、規約の定めるところにより、何を経て交代させることができるでしょうか？	代議員会の議決。	Wrong

Figure A.1 (continued)

39	基金は、事業の継続の不能を理由に解散しようとするときは、誰の認可を受けなければならないでしょうか？	厚生労働大臣。	基金は、厚生労働大臣の認可を受けなければならない。
40	基金が代議員の定数の四分の三以上の多数による代議員会の議決により解散したときは、原則として誰がその清算人となりますか？	理事。	Wrong
41	清算人が欠けたため損害を生ずるおそれがあるときは、厚生労働大臣が清算人を選任しますが、その場合において基金は何を負担するのでしょうか？	清算人の職務の執行に要する費用。	Wrong
42	清算人は、債権の取立て及び債務の弁済を行うためにどんな行為をすることができますか？	必要な一切の行為をすることができます。	清算人は、前項各号に掲げる職務を行うために必要な一切の行為をすることができる。
43	清算人は、その就職の日から二箇月以内に、少なくとも三回の公告をもつて、債権者に対し、債権の申出の催告をしなければなりませんが、その広告はどこに掲載してするのでしょうか？	官報。	官報に掲載してする。
44	清算人は、少なくとも三回の公告をもつて、債権者に対し、一定の期間内にその債権の申出をすべき旨の催告をしなければならないが、その広告に付記しなければならない事項は何か？	債権者が一定の期間内に申出をしないときは清算から除斥されるべき旨。	前項の公告には、債権者がその期間内に申出をしないときは清算から除斥されるべき旨を付記しなければならない
45	基金による交付の申出に係る現価相当額の計算については、何で定められているのか？	政令で定められている。	政令で定める。
46	連合会は、基金による現価相当額の交付の申出があつたときは、どのような対応をしなくてはならないのでしょうか？	交付の申出を拒絶してはならない。	連合会は、これを拒絶してはならない。
47	保険料その他国民年金法の規定による徴収金を滞納する者に対して、厚生労働大臣が期限を指定してする督促は、何の効力を有するか？	時効中断の効力を有する。	Wrong
48	保険料以外の国民年金法の規定による徴収金を滞納する者に対して、厚生労働大臣が期限を指定して督促をなしたときは、厚生労働大臣は、原則として徴収金額に納期限の翌日から徴収金完納又は財産差押の日の前日までの期間の日数に応じ、年何パーセントの割合を乗じて計算した延滞金を徴収するのか？	十四・六パーセント。	年七・三パーセント）の割合を乗じて計算した延滞金を徴収する。 Wrong (True, if we read the whole paragraph, which contains the above extracted answer)
49	被保険者又は被保険者であつた者が厚生労働大臣の承認を受けて保険料を追納する場合において、追納すべき額は当該追納に係る期間の各月の保険料の額に何を加算した額でしょうか？	政令で定める額。	当該追納に係る期間の各月の保険料の額に政令で定める額を加算した額とする。
50	厚生労働大臣は、被保険者の資格を取得した旨の報告を受けたとき、原則として当該被保険者について国民年金手帳を作成しなければなりませんが、厚生労働大臣に対し、その報告をするのは誰でしょうか？	市町村長。	厚生労働大臣は、前条第四項の規定により被保険者の資格を取得した旨の報告を受けたとき、又は同条第五項の規定により第三号被保険者の資格の取得に関する届出を受理したときは、当該被保険者について国民年金手帳を作成し、 And the referent to A12P4 市町村長は、第一項又は第二項の規定による届出を受理したときは、厚生労働省令の定めるところにより、厚生労働大臣にこれを報告しなければならない。

Figure A.1 (continued)

Bibliography

[1] A. Agarwal, H. Raghavan, K. Subbian, P. Melville, R.D. Lawrence, and D.C. Gondek. Learning to rank for robust question answering. In *Proceedings of the 21st ACM international conference on Information and knowledge management (CIKM)*, pages 833–842, 2012.

[2] R.D Anne, O. Yilmazel, and E.D. Liddy. Evaluation of restricted domain question-answering systems. In *Proceedings of Workshop on Question Answering in Restricted Domains. 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, pages 2–7, 2004.

[3] G. Attardi, S.D. Rossi, and M. Simi. TANL-1: Coreference resolution by parse analysis and similarity clustering. In *Proceedings of SemEval-2*, pages 108–111, 2010.

[4] S. Azzam, K. Humphreys, and R. Gaizauskas. Using coreference chains for text summarization. In *Proceeding CorefApp '99 Proceedings of the Workshop on Coreference and its Applications*, pages 77–84, 1999.

[5] N.X Bach, N.L. Minh, and A. Shimazu. Recognition of requisite part and effectuation part in law sentences. In *Proceedings of the 23rd International Conference on the Computer Processing of Oriental Languages (ICCPOL)*, pages 29–34, 2010.

[6] N.X. Bach, N.L. N.L. Minh, and A. Shimazu. RRE task: The task of recognition of requisite part and effectuation part in law sentences. *Journal of IJCPOL*, 23(2): 109–130, 2010.

[7] N.X Bach, N.L. Minh, T.T. Oanh, and A. Shimazu. A two-phase framework for learning logical structures of paragraphs in legal articles. *Journal of ACM Transactions on Asian Language Information Processing (ACM TALIP)*, 12(1, article no 3):1–32, 2013.

[8] A. Bagga and B. Baldwin. Algorithms for scoring coreference chains. In *Proceedings of LREC Workshop on Linguistic coreference*, pages 563–566, 1998.

[9] L.E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistic*, 37:1554–1563, 1966.

[10] S.J. Benson and J.J. More. A limited-memory variable-metric method for bound-constrained minimization. In *Preprint ANL/MCS-P909-0901*, 2001.

[11] A.L Berger, V.J.D Pietra, and S.A.D Pietra. A maximum entropy approach to natural language processing. *Journal of Computational Linguistics*, 22:39–71, 1996.

[12] A. Bolioli, L. Dini, P. Mercatali, and F. Romano. For the automated mark-up of Italian legislative texts in XML. In *Proceedings of International on Legal Knowledge and Information Systems (Jurix)*, pages 21–30, 2002.

[13] E. Breck, J. Burger, L. Ferro, D. House, M. Light, and I. Mani. A sys called Qanda. In *Proceedings of the 8th Text Retrieval Conference (TREC-8)*, pages 499–506, 1999.

[14] S. Brennan, M. Friedman, and C. Pollard. A Centering approach to pronoun. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 115–162, 1987.

[15] P.F. Brown, P.V. deSouza, R.L Mercer, V.J.D. Pietra, and J.C. Lai. Class-based n-gram models of natural language. *Journal of Computational Linguistics*, 18(4): 467–479, 1992.

[16] Chris J.C. Burges and Bernhard Schlkopf. Improving the accuracy and speed of support vector machines. In *Advances in Neural Information Processing Systems 9*, pages 375–381. MIT Press, 1997.

[17] Christopher J. C. Burges. Advances in kernel methods. chapter Geometry and invariance in kernel based methods, pages 89–116. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-19416-3. URL `http://dl.acm.org/citation.cfm?id=299094.299100`.

[18] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[19] Z. Cao, T. Qin, T.Y. Liu, M.F. Tsai, and H. Li. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 129–136, 2007.

[20] C.C. Chang and C.J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[21] C. Cortes and V. Vapnik. Support-vector networks. *Journal of Machine Learning*, 20:273–297, 1995.

[22] A. Culotta and A. McCallum. Joint deduplication of multiple record types in relational data. In *Proceedings of the 14th ACM international conference on Information and knowledge management (CIKM)*, pages 257–258, 2005.

[23] W. Daelemans, J. Zavrel, K. Sloot, and A. Bosch. TiMBL: Tilburg memory based learner version 6.1 reference guide. Technical Report ILK 07-07, Tilburg University, 2007.

[24] J.N. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *The Annals of Mathematical Statistics*, 43(5):1470–1480, 1972.

[25] P. Denis and J. Baldridge. A ranking approach to pronoun resolution. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1588–1593, 2007.

[26] P. Denis and J. Baldridge. Specialized models and ranking for coreference resolution. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 660–669, 2008.

[27] S. Dipper and H. Zinsmeister. Annotating abstract anaphora. *Journal of Language Resources and Evaluation*, 46(1):37–52, 2012.

[28] H. Doan-Nguyen and L. Kosseim. The problem of precision in restricted-domain question-answering. In *Proceedings of Workshop on Question Answering in Restricted Domains. 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, pages 8–15, 2004.

[29] C. Dyer. Using a maximum entropy model to build segmentation lattices for MT. In *Proceedings of North American Chapter of the Association for Computational Linguistics - Human Language Technologies(NAACL-HLT)*, pages 406–414, 2009.

[30] M. Eckert and M. Strube. Dialogue acts, synchronizing units, and anaphora resolution. *Journal of Semantics*, 17:51–89, 2000.

[31] Jenny Rose Finkel and Christopher D. Manning. Hierarchical joint learning: Improving joint parsing and named entity recognition with non-jointly labeled data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2010. URL `pubs/hier-joint.pdf`.

[32] J.R. Finkel and C.D. Manning. Hierarchical joint learning: Improving joint parsing and named entity recognition with non-jointly labeled data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 720–728, 2010.

[33] R. Gaizauskas and K. Humphreys. A combined IR/NLP approach to question answering against large text collections. In *Proceedings of the 6th Content-Based Multimedia Information Access Conference (RIAO-2000)*, pages 1288–1304, 2000.

[34] N. Ge, J. Hale, and E. Charniak. A statistical approach to anaphora resolution. In *In Proceedings of the Sixth Workshop on Very Large Corpora*, pages 161–170, 1998.

[35] B. Hachey, W. Radford, and J.R. Curran. Graph-based named entity linking with Wikipedia. In *Proceedings of the 12th International Conference on Web Information System Engineering (WISE)*, pages 213–226, 2011.

[36] B. Hachey, W. Radford, J. Nothman, M. Honnibal, and J.R. Curran. Evaluating entity linking with Wikipedia. *Journal of Artificial Intelligence*, 194:130–150, 2013.

[37] A. Haghighi and D. Klein. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 1152–1161, 2009.

[38] S. Hartrumpf, I. Glckner, and J. Leveling. Coreference resolution for questions and answer merging by validation. In *Advances in Multilingual and Multimodal Information Retrieval, Lecture Notes in Computer Science*, pages 269–272, 2008.

[39] I. Hendrickx, G. Bouma, W. Daelemans, and V. Hoste. COREA: Coreference resolution for extracting answers for Dutch. *Journal of Essential Speech and Language Technology for Dutch, Theory and Applications of Natural Language Processing*, pages 115–128, 2013.

[40] J. Heng and G. Ralph. Applying coreference to improve name recognition. In *Proceedings of ACL 2004: Workshop on Reference Resolution and its Applications*, pages 32–39, 2004.

[41] R. Herbrich, T. Graepel, and K. Obermayer. *Large margin rank boundaries for ordinal regression*, chapter 7, pages 115–132. MIT Press, 2000.

[42] J. Hobbs. *Resolving pronoun references*. Morgan Kaufmann Publishers, 1986.

[43] R. Iida, K. Inui, H. Takamura, and Y. Matsumoto. Incorporating contextual cues in trainable models for coreference resolution. In *Proceedings of EACL Workshop on the computational Treatment of anaphora*, pages 23–30, 2003.

[44] R. Jin, H. Valizadegan, and H. Li. Ranking refinement and its application to information retrieval. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 397–406, New York, NY, USA, 2008. ACM.

[45] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining (KDD)*, pages 133–142, 2002.

[46] T. Joachims. Training linear SVMs in linear time. In *Proceedings of ACM Conference on Knowledge Discovery and Data Mining (KDD)*, pages 217–226, 2006.

[47] D. Jurafsky and J.H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall Series in Artificial Intelligence, 2nd edition, 2009.

[48] M. Kahng, S. Lee, and S.G. Lee. Ranking in context-aware recommender systems. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, pages 65–66, New York, NY, USA, 2011. ACM.

[49] T. Katayama. Legal Engineering - an engineering approach to laws in e-society age. In *Proceedings of International Workshop on Juris-informatics (JURISIN)*, 2007.

[50] T. Katayama. The curent status of the art of the 21st COE programs in the information sciences field. verifiable and evolvable e-society - realization of trustworthy e-society by computer science - (in japanese). *Journal of Information Processing Society of Japan*, 46(5):515–521, 2010.

[51] T. Katayama, A. Shimazu, S. Tojo, K. Futatsugi, and K. Ochimizu. e-society and legal engineering (in japanese). *Journal of the Japanese Society for Artificial Intelligence*, 23(4):529–536, 2008.

[52] H. Kobdani and H. Schutze. SUCRE: Modular system for coreference resolution. In *Proceedings of SemEval-2*, pages 92–95, 2010.

[53] T. Kudo and Y. Matsumoto. Japanese dependency analysis using cascaded chunking. In *Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops) (CoNLL 2002)*, pages 63–69, 2002.

[54] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 282–289, 2001.

[55] S. Lapping and H.J. Leass. An algorithm for pronominal anaphora resolution. *Journal of Computational Linguistics*, 20:535–561, 1994.

[56] Y.H. Lee, M.Y. Kim, and J.H Lee. Chunking using conditional random fields in Korean texts. In *Proceedings of International Joint Conference on Natural Language Processing (IJCNLP)*, pages 155–164, 2005.

[57] H. Li. *Learning to rank for information retrieval and natural language processing.* Synthesis leactures on human language technologies, Graeme Hirst, Series Editor, 2011.

[58] P. Liang. Semi-supervised learning for natural language. In *Master's thesis, Massachusetts Institute of Technology*, 2005.

[59] T.Y. Liu. *Learning to rank for information retrieval.* Springer Publisher, 2011.

[60] D. Ludtke and S. Sato. Fast base NP chunking with decision trees experiments on different POS tag settings. In *Proceedings of Conferences on Computational Linguistics and Natural Language Processing (CICLing)*, pages 139–150, 2003.

[61] X. Luo. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP)*, pages 25–32, 2005.

[62] X. Luo, A. Ittycheria, H. Jing, N. Kambhatla, and S. Roukos. A mention-synchronous coreference resolution algorithm based on the Bell tree. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 135–142, 2004.

[63] M. Ciaramita M. Surdeanu and H. Zaragoza. Learning to rank answers on large online QA collections. In *Proceedings of the 46th annual meeting of the Association for Computational Linguistics: Human Language Technology (ACL-HLT)*, pages 719–727, 2008.

[64] E.D. Maat, R. Winkels, and T.V. Engers. Automated detection of reference structures in law. In *Proceedings of International on Legal Knowledge and Information Systems (Jurix)*, pages 41–50, 2006.

[65] C.D Manning, P. Raghana, and H. Schutze. *Introduction to Information Retrieval.* Cambridge University Press, 2008.

[66] A. McCallum and B. Wellner. Conditional models of identity uncertainty with application to proper noun coreference. In *Neural Information Processing Systems (NIPS)*, 2004.

[67] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, pages 591–598, 2000.

[68] M.G. Mercedes, D.L.F. Pablo, and V. Dámaso-Javier. Reference extraction and resolution for legal texts. In *Proceedings of PReMI*, pages 218–221, 2005.

[69] A. Monroy, H. Calvo, and A. Gelbukh. NLP for shallow question answering of legal documents using graphs. In *Proceedings of the 10th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, pages 498–508, 2009.

[70] T.S. Morton. Using coreference for question answering. In *Proceeding CorefApp '99 Proceedings of the Workshop on Coreference and its Applications*, pages 85–89, 1999.

[71] T.S. Morton. Using coreference to improve passage retrieval for question answering. Technical report, In Proceedings of the AAAI Fall Symposium on Question Answering Systems, 1999.

[72] T.S. Morton. Using coreference in question answering. In *Proceedings of the ACL99 W orkshop on Conference and its Applications*, pages 85–89, 1999.

[73] C. Navarretta. Resolving individual and abstract anaphora in texts and dialogues. In *Proceedings of the 20th international conference on Computational Linguistics (COLING)*, 2004.

[74] V. Ng. Supervised ranking for pronoun resolution: Some recent improvements. In *Proceedings of National Conference on Artificial Intelligence (AAAI)*, pages 1081–1086, 2005.

[75] V. Ng. Semantic class induction and co-reference resolution. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 536–543, 2007.

[76] V. Ng. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1396–1411, 2010.

[77] V. Ng and V. Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 104–111, 2002.

[78] V. Ng and V. Cardie. Identifying anaphoric and non-anaphoric noun phrases to improve coreference resolution. In *Proceedings of International Conference on Computational Linguistics(COLING)*, pages 730–736, 2002.

[79] J. Nocedal. Updating Quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.

[80] M. Novak. Utilization of anaphora in machine translation. In *Proceedings of WDS*, pages 155–160, 2011.

[81] J.H. Paik. A novel TF-IDF weighting scheme for effective ranking. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval (SIGIR)*, pages 343–352, 2013.

[82] M. Palmirani, R. Brighi, and M. Massini. Automated extraction of normative references in legal texts. In *Proceedings of International Conference on Artificial Intelligence and Law (ICAIL)*, pages 105–106, 2003.

[83] G. Paltoglo and M. Thelwall. A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1386–1395, 2010.

[84] M. Paul and E. Sumita. Utilization of coreferences for the translation of utterances containing anaphoric expressions. In *Proceedings of Pacific Rim International Conference on Artificial Intelligence (PRICAI)*, pages 820–820, 2000.

[85] Q. Paulo and I.P. Rodrigues. A question-answering system for portuguese juridical documents. In *Proceedings of the 10th international conference on Artificial Intelligence and Law (ICAIL)*, pages 256–257, 2005.

[86] A. Penas, P. Forner, R. Sutcliffe, A. Rodrigo, C. Forascu, I. Alegria, D. Giampiccolo, N. Moreau, and P. Osenova. Overview of respubliqa 2009: Question answering evaluation over european legislation. In *CLEF 2009 Workshop, Part 1, LNCS 6241*, pages 174–196, 2010.

[87] S. Peral, M. Palomar, and A. Ferrandez. Coreference-oriented interlingual slot structure and machine translation. In *Proceedings CorefApp '99 Proceedings of the Workshop on Coreference and its Applications*, pages 69–76, 1999.

[88] S.D Pietra, V.D Pietra, and J. Lafferty. Inducing features of random fields. Technical report, CMU-CS-95-144, Carnegie Mellon University, 1995.

[89] D. Pinto, A. McCallum, X. Wei, and W.B Croft. Table extraction using conditional random fields. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 235–242, 2003.

[90] E. Pitler, S. Bergsma, D. Lin, and K. Church. Using web-scale n-grams to improve base NP parsing performance. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 886–894, 2010.

[91] S.P. Ponzetto and M. Strube. Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proceedings of Human Language Technologies: Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 192–199, 2006.

[92] J.R. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1 edition, January 1993.

[93] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.

[94] F. Radlinski and T. Joachims. Query chains: Learning to rank from implicit feedback. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, KDD '05, pages 239–248, New York, NY, USA, 2005.

[95] A. Rahman and V. Ng. Supervised models for coreference resolution. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pages 968–977, 2009.

[96] A. Rahman and V. Ng. Ensemble-based coreference resolution. In *Proceedings of International Joint Conference on Artifical Intelligence (IJCAI)*, pages 1884–1889, 2011.

[97] A. Ratnaparkhi. A simple introduction to maximum entropy models for natural language processing. Technical report, Institute for Research in Cognitive Science, University of Pennsylvania, 1997.

[98] M. Recasens and E. Hovy. BLANC: Implementing the Rand Index for coreference evaluation. *Natural Language Engineering*, 17(4):485–510, 2011.

[99] M. Recasens, L. Marquez, L. Sapena, M. Marti, M. Taule, V. Hoste, M. Poesio, , and Y. Versley. Semeval-2010 task 1: Co-reference resolution in multiple languages. In *Proceedings of International Workshop on Semantic Evaluation*, pages 1–8, 2010.

[100] S. Robertson and S. Walker. On relevance weights with little relevance information. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 16–24, 1997.

[101] S. Robertson and S. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Journal of Foundations and Trends in Information Retrieval*, 3:333–389, 2009.

[102] E. Sapena, L. Padro, and J. Turmo. RelaxCor: A global relaxation labeling approach to coreference resolution for the SemEval-2 coreference task. In *Proceedings of SemEval-2*, pages 88–91, 2010.

[103] F. Sha and F. Pereira. Shallow parsing with conditional random fields. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 213–220, 2003.

[104] W.M. Soon, D.C.Y. Lim, and H.T. Ng. A machine learning approach to co-reference resolution of noun phrases. *In Journal of Computational Linguistics*, 27(4):521–544, 2001.

[105] J. Steinberger, M. Poesio, M.A. Kabadjov, and K. Jeek. Two uses of anaphora resolution in summarization. *Information Processing and Management*, 43(6):1663–1680, 2007.

[106] R. Stuckardt. Coreference-based summarization and question answering: A case for high precision anaphor resolution. In *Proceedings of International Symposium on Reference Resolution*, pages 33–41, 2003.

[107] Charles Sutton and Andrew McCallum. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[108] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Journal of Foundations and Trends in Machine Learning*, 4(4):267–373, 2012.

[109] K. Takano, M. Nakamura, Y. Oyama, and A. Shimazu. Semantic analysis of paragraphs consisting of multiple sentences - towards development of a logical formulation system. In *Proceedings of the 23rd International Conference on Legal Knowledge and Information Systems (JURIX10)*, pages 117–126, 2010.

[110] K. Tanaka, I. Kawazoe, and H. Narita. Standard structure of legal provisions for the legal knowledge processing by natural language (in Japanese). In *Res. rep. on Natural Language Processing, IPSJ*, pages 79–86, 1993.

[111] K. Tomura. Study on question answering system for laws. Technical report, School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), 2013.

[112] K. Toutanova and C.D. Manning. Enriching the knowledge sources used in a maximum entropy Part-of-Speech tagger. In *Proceedings of J. SIGDAT Conf. on Empirical Methods in NLP and Very Large Corpora (EMNLP/VLC-2000)*, pages 63–70, 2000.

[113] O.T Tran, C.A. Le, T.Q. Ha, and Q.H. Le. An experimental study on Vietnamese POS tagging. In *Proceedings of International Conference on Asia Language Processing (IALP)*, pages 23–27, 2009.

[114] O.T. Tran, M.L. Nguyen, and A. Shimazu. Reference resolution in legal texts. In *Proceedings of ICAIL*, pages 101–110, 2013.

[115] V.N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[116] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. A model-theoretic coreference scoring scheme. In *Proceedings of MUC-6*, pages 45–52, 1995.

[117] H.M. Wallach. Efficient training of conditional random fields. Master's thesis, University of Edinburgh, 2002.

[118] Mengqiu Wang, Wanxiang Che, and Christopher D. Manning. Joint word alignment and bilingual named entity recognition using dual decomposition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2013. URL `http://nlp.stanford.edu/pubs/wang-etal-acl13.pdf`.

[119] T. Watanabe. Optimized online rank learning for machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 253–262, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[120] C. Whidden. Utilizing automatic coreference resolution with the Jellyfish question answering system. Technical report, Dalhousie FCS Technical Report, 2007.

[121] R. Witte and S. Bergler. Fuzzy coreference resolution for summarization. In *Proceedings of 2003 International Symposium on Reference Resolution and Its Applications to Question Answering and Summarization (ARQAS)*, pages 43–50, 2003.

[122] F. Xia, T.Y. Liu, W. Wang, and H. Li. Listwise approach to learning to rank: Theory and algorithm. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 1192–1199, 2008.

[123] J. Xu and H. Li. AdaRank: A boosting algorithm for information retrieval. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 391–398, 2007.

[124] Z. Xu, X. Qian, Y. Zhang, and Y. Zhou. CRF-based hybrid model for word segmentation, NER and even POS tagging. In *Proceedings of International Joint Conference on Natural Language Processing (IJCNLP)*, pages 167–170, 2008.

[125] X. Yang, G. Zhou, J. Su, and C.L. Tan. Coreference resolution using competitive learning approach. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 176–183, 2003.

[126] X. Yang, J. Su, G. Zhou, and C.L. Tan. An NP-cluster based approach to coreference resolution. In *Proceedings of International Conference on Computational Linguistics(COLING)*, pages 226–232, 2004.

[127] W. Zhang, T. Yoshida, and X. Tang. A comparative study of TF*IDF, LSI and multi-words for text classification. *Expert Systems with Applications: An International Journal*, 38:2758–2765, 2011.

[128] D. Zhekova and S. Kubler. UBIU: A language-independent system for coreference resolution. In *Proceedings of SemEval-2*, pages 96–99, 2010.

Publications

Journal Articles

[1] **Oanh Thi Tran**, Bach Xuan Ngo, Minh Le Nguyen, Akira Shimazu. Automated Reference Resolution in Legal Texts, 2013. *Journal of Artificial Intelligence and Law*, DOI:10.1007/s10506-013-9149-8, 22(1), 2014.

[2] **Oanh Thi Tran**, Bach Xuan Ngo, Minh Le Nguyen, Akira Shimazu. An Empirical Study on a Listwise Approach to Coreference Resolution using Learning-to-rank. (submitted to the Journal of Knowledge-Based Systems)

[3] Ngo Xuan Bach, Nguyen Le Minh, **Tran Thi Oanh**, Akira Shimazu. A Two-Phase Framework for Learning Logical Structures of Paragraphs in Legal Articles. *ACM Transactions on Asian Language Information Processing (ACM TALIP)*, 12(1), article 3, 2013.

Referred Conference Papers

[4] **Oanh Thi Tran**, Bach Xuan Ngo, Minh Le Nguyen, Akira Shimazu. Answering Legal Questions by Mining References Information. In *Post-Proceedings of the 7th International Workshop on Juris-informatics (JURISIN)*, Lecture Notes in AI, Yokohama, Japan, 2013.

[5] **Oanh Thi Tran**, Bach Xuan Ngo, Minh Le Nguyen, Akira Shimazu. Reference Resolution in Japanese Legal Texts at Passage Levels. In *Proceedings of the 5th International Conference on Knowledge and Systems Engineering (KSE)*, Springer-Verlag, pages 237–249 , 2013.

[6] **Oanh Thi Tran**, Minh Le Nguyen, Akira Shimazu. Reference Resolution in Legal Texts. In *Proceedings of the 14th International Conference on Artificial Intelligent and Law (ICAIL)*, pages 101-110, Rome, Italy, June 2013. (***Best student paper award***)

[7] **Oanh Thi Tran**, Bach Xuan Ngo, Minh Le Nguyen, Akira Shimazu. A Listwise Approach to Coreference Resolution in Multiple Languages. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation (PACLIC)*, pages 400–409, 2011.

[8] Ngo Xuan Bach, Nguyen Le Minh, **Tran Thi Oanh**, Akira Shimazu. Learning Logical Structures of Paragraphs in Legal Articles. In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 20–28, 2011.