Decision Support Systems 50 (2011) 743–754

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Mining comparative opinions from customer reviews for Competitive Intelligence

Kaiquan Xu [a,*], Stephen Shaoyi Liao [a], Jiexun Li [b], Yuxia Song [a]

[a] Department of Information Systems, City University of Hong Kong, Kowloon, Hong Kong SAR, Hong Kong
[b] College of Information Science and Technology, Drexel University, Philadelphia, PA 19104, USA

ARTICLE INFO

Available online 19 August 2010

Keywords:
Opinion mining
Enterprise risk management
Competitive Intelligence

ABSTRACT

Competitive Intelligence is one of the key factors for enterprise risk management and decision support. However, the functions of Competitive Intelligence are often greatly restricted by the lack of sufficient information sources about the competitors. With the emergence of Web 2.0, the large numbers of customer-generated product reviews often contain information about competitors and have become a new source of mining Competitive Intelligence. In this study, we proposed a novel graphical model to extract and visualize comparative relations between products from customer reviews, with the interdependencies among relations taken into consideration, to help enterprises discover potential risks and further design new products and marketing strategies. Our experiments on a corpus of Amazon customer reviews show that our proposed method can extract comparative relations more accurately than the benchmark methods. Furthermore, this study opens a door to analyzing the rich consumer-generated data for enterprise risk management.

1. Introduction

Competitive Intelligence (CI) involves the early identification of potential risks and opportunities by gathering and analyzing information about the environment to support managers in making strategic decisions for an enterprise [33]. Most firms realize the importance of CI in enterprise risk management and decision support, and invest a large amount of money in CI. A survey from the American Futures Group consulting firm indicates that 82% of large enterprises and over 90% of the Forbes top 500 global firms adopt CI for risk management and decisions. By the end of the 20th century, the overall production value of CI industry had reached 70 billion U.S. dollars [23].

In order to identify potential risks, it is important for companies to collect and analyze information about their competitors' products and plans. Based on such information, a company can learn the relative weaknesses and strengths of its own products, and can then design new pointed products and campaigns to countervail those of its competitors. Traditionally, information about competitors has mainly come from press releases, such as analyst reports and trade journals, and recently also from competitors' websites and news sites. Unfortunately, such information is mostly generated by the company that produces the product. Therefore, the amount of available information is limited and its objectivity is questionable. The lack of sufficient and reliable information sources about competitors greatly restricts the capability of CI.

With the emergence of Web 2.0, an increasing number of customers now have opportunities to directly express their opinions and sentiments regarding products through various channels, such as online shopping sites, blogs, social network sites, forums, and so forth. These opinion data, coming directly from customers, become a natural information source for CI. There are some existing studies on mining customer opinions [6,7,27,31,34]. However, these studies mainly focus on identifying customers' sentiment polarities toward products. The most important problem in CI—i.e., collecting and analyzing the competitors' information to identify potential risks as early as possible and plan appropriate strategies—has not been well studied.

Customer reviews are often a rich source of comparison opinions. Users usually prefer to compare several competitive products with similar functions, for example,

Nokia N95 has a stronger signal than iPhone.

The iPhone has better looks, but a much higher price than the BB Curve.

Compared with the v3, this V8 has a bigger body, and it has a much worse keyboard than Nokia E71.

These comparison opinions are precious information sources for identifying the relative strengths and weaknesses of products, analyzing the enterprise risk and threats from competitors, and further designing new products and business strategies.

Mining such comparison opinions is a non-trivial task due to the large amount of customer reviews and their informal style. In this

* Corresponding author.
E-mail address: kaiquan.xu@student.cityu.edu.hk (K. Xu).

0167-9236/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2010.08.021

paper, we propose a novel approach to extracting product comparative relations from customer reviews, and display the results as comparative relation maps for decision support in enterprise risk management.

The remainder of this paper is organized as follows: Section 2 reviews the related work in comparative opinion mining. Section 3 introduces our overall approach of comparative relation extraction. Section 4 introduces a novel graphical model we propose for comparative relation extraction. Section 5 presents our experiments that evaluate the proposed relation extraction approach. Section 6 concludes our study and discusses some future directions for research.

2. Related work

2.1. Sentiment analysis of user opinions

Much research exists on sentiment analysis of user opinion data [6,7,27,31,34], which mainly judges the polarities of user reviews. In these studies, sentiment analysis is often conducted at one of three levels: the document level, sentence level, or attribute level. Sentiment analysis at the document level classifies reviews into the types of polarities—positive, negative, or neutral—based on the overall sentiments in the reviews. A number of machine learning techniques have been adopted to classify the reviews [32]. Abbasi and Chen et al. propose the sentiment analysis methodologies for classification of Web forum opinions in multiple languages [1]. Sentiment analysis at the sentence level mainly focuses on identifying subjective sentences and judging their polarities. Most of these studies adopted the machine learning methods [42,47]. Sentiment analysis at both the document level and sentence level has been too coarse to determine precisely what users like or dislike. In order to address this problem, sentiment analysis at the attribute level is aimed at extracting opinions on products' specific attributes from reviews. In [17], Part Of Speech (POS) tag sequence rules were used to extract product attributes, and then the polarities of opinion phrases on the attributes were judged based on the context information. For the sentiment analysis, various features can be used. Term presence has been more effective than term frequency in classifying the polarities of documents, and the positions of terms also have had an important influence on sentiment analysis [32]. The POS tags of words, such as adjectives and adverbs, have been good indicators for the subjectivity detection and sentiment polarity classification [2,40]. The syntax features (such as dependency tree) outperform the bag-of-words for the sentiment polarity classification in some situations [24]. The interactions between topic and sentiment play an important role in sentiment analysis [13]. The techniques for sentiment analysis are mainly classified into two categories: unsupervised approaches and supervised approaches. The unsupervised approaches usually create a sentiment lexicon and determine the polarities by counting the positive and negative phrases [15,40]. The supervised approaches use the labeled data to train some classifiers (such as Naive Bayes, Maximum Entropy, or Support Vector Machine) to predict the unlabeled data [32,42,47]. Other important research on sentiment analysis includes identifying the sentiment target/topic and the opinion holder. The purpose of identifying the sentiment target/topic is to discover subjectivity and sentiment sentences [43,45,46], and the existing research proposes various linguistic clues for this task. For the task of identifying the opinion holders associated with particular opinions [3,22], the semantic parsing techniques are proposed.

In addition, several systems [28,45,46] have been developed to automatically analyze customer reviews, mine opinions toward a product or attribute, and visually show the mined information for aiding users' decision making. Some systems [28] can aggregate opinions about certain attributes of several competing products and help users compare their pros and cons. However, these existing systems only mine general user opinions, which can be biased in acquiring competitors' information and identifying the potential operational risks. Unlike these systems, our study focuses on mining users' comparative opinions from product reviews. Because these comparative opinions can better reflect customers' preferences on competitive products, they should be more effective in tracing information regarding competitors and supporting enterprise risk management.

As noted, these studies mainly focused on judging customers' sentiment polarities toward products. However, few studies have focused on extracting the sentiment polarities of user opinions on comparisons of competitive products. Usually, this type of sentiment polarity is more important for enterprises to learn, so as to discover their products' weaknesses and design pointed products.

2.2. Relation extraction

Another closely linked research area is relation extraction, which detects if there exists a specific relation between entities, such as *work_in (Tom, IBM)* (meaning *Tom works in IBM*). A number of methods currently exist for relation extraction. Some of these methods are based on rules/templates, and others formulate the relation extraction as a classification problem and use various classification techniques to resolve it. In the rule-based methods, the extraction rules are defined manually [9] or learned from large annotated training corpora [14]. The classification-based methods can be divided into two categories: feature-based methods [21] and kernel-based methods [4,5]. Feature-based methods define the feature set (such as words, POS tags, entity types, path in parse tree, etc.), and represent the examples using these features to train classifiers. Usually, the computing complexity of this kind of method is relatively low, while the choice of features is intuitive and difficult. The kernel-based methods structurally represent examples and define kernels to compute the similarities in a high-dimension space implicitly. For example, in [8] and [48], the shallow parse tree kernels and the dependency tree kernel were used separately. In [26], a trace kernel was incorporated with the tree kernel to capture richer contextual information for biomedical relation extraction. This kind of methods does not need to define a feature set, but the computation complexity is relatively high.

Comparative relation extraction differs from the existing relation extraction in two important ways: 1) the comparative relation is a higher-order relation. That is, one comparative relation contains four entities/arguments, while the existing relation extraction methods mainly address relations with two entities. 2) Comparative relation extraction not only detects if the comparative relations occur or not, but also recognizes their directions. That is, the extraction task needs to indicate that product *A* is better than product *B* on a particular attribute, or the inverse. In contrast, the existing relation extraction methods mainly detect merely the occurrences of relations. These two characteristics bring particular challenges to comparative relation extraction: Involvement of multiple entities means that the long-range features need to be captured for better extraction; recognizing the direction of the relation makes the comparative relation extraction become a multi-class classification problem, unlike the existing relation extraction tasks, which were typically binary-class classification problems.

The only work up to now on extracting the comparative relations from customer opinions is [18]. Their work only identified the comparative sentences and extracted the relation items. In their research, the comparative relation included two product entities, one attribute entity, and keywords expressing the comparative relation. The comparative relations were classified into three types: "*non-equal gradable*", "*same*", and "*superlative*" (For the "*non-equal gradable*" comparative relation, they did not differentiate the directions.). In addition, a very complicated process was adopted: First, the candidate comparative sentences were filtered out using specific keywords. Then, a Bayesian classifier was trained using the sequence rules built based on the training examples and manual compilation, for detecting comparative sentences. Following that, the Support Vector Machine (SVM) classifier was used to classify the "*non-equal gradable*", "*same*", and "*superlative*" relations. The researchers assumed that there was only one comparative relation in a sentence, and sequence rules were used to extract the comparative relation items.

Their method has several limitations: First, the assumption of only one relation per sentence is violated frequently because users often prefer to compare several products, or different attributes of competitive products, in one sentence. Second, the requirement of manually compiling rules makes this method difficult to adapt to new domains. In addition, the rule-based method usually has good precision, but the recall is often low [35]. Third, their method cannot recognize the directions of the "*non-equal gradable*" comparative relations, which makes it insufficient to judge the sentiment polarities of customers on the competitive products.

2.3. Graphical model

Graphical models are a powerful tool to model the complicated problems in an intuitive way [10]. They have been applied to classify semantic relations in bioscience texts [36]. Graphical models can be categorized into directed graphical model (Bayesian networks) and undirected graphical model. The recently emerging Conditional Random Fields (CRF) [25] is an undirected graphical model. Compared with the Bayesian networks, CRF directly models the conditional probability distribution of the output given the input, so it can exploit the rich and global features of the inputs without representing the dependencies in the inputs. Also CRF needs to estimate fewer parameters than the Bayesian networks, so it has excellent performance when the training sample is small. The comparative relation extraction involves multiple entities and long-range dependencies, and needs to capture rich features from the inputs, so CRF is an ideal tool to use for it.

CRF encodes a conditional probability distribution using a given set of features. A typical and commonly used model is the linear-chain CRF with one level (see graphical representation in Fig. 1), which defines the conditional probability of a label sequence $\mathbf{y}$ given the observed sequence $\mathbf{x}$ as

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(x)} \prod_{t=1}^{T} \Psi_t(y_t, y_{t-1}, \mathbf{x}_t)$$

where $Z(\mathbf{x})$ is a normalization function, $\Psi_t(y_t, y_{t-1}, \mathbf{x}_t)$ is the factor, and can be parameterized as

$$\Psi_t(y_t, y_{t-1}, \mathbf{x}_t) = \exp\left\{\sum_k \lambda_k f_k(y_t, y_{t-1}, \mathbf{x}_t)\right\}$$

where λ_k is the parameter to represent the weights of features and $f_k(y_t, y_{t-1}, \mathbf{x}_t)$ is the feature function.

The linear-chain CRF with one level is often used for sequence labeling problems, such as POS tag labeling and phrase chunking. However, the comparative relation extraction needs to capture the features from both the entity and the word levels, so the linear-chain CRF with one level is not enough. In addition, in existing CRF models, the dependencies between nodes are fixed and always exist, like the dependencies between the neighboring y nodes in the linear-chain CRF. However, for the comparative relation extraction, there exist dependencies between the relations in one sentence in some situations, but these dependencies are unfixed and conditional on the inputs. Hence, the existing CRF models are not very appropriate or useful for the comparative relation extraction.

2.4. Summary

In summary, extracting the comparative relations from customer comparison opinions is very important to find out the sentiment polarities of customers on competitive products. The existing relation extraction methods cannot handle the high-order relation extraction problem very well, and cannot recognize the directions of the comparative relations. The CRF model is a potentially powerful tool for modeling the complicated dependencies in comparative relation extraction and recognizing the directions of the relations. However, the existing CRF models cannot capture the features from entity and word levels, and cannot model the unfixed dependencies between the relations. Our study is aimed at overcoming these limitations and developing a new CRF model to better extract comparative relations from customer reviews.

3. Overall process of mining comparative opinions

3.1. Problem formulation

Most comparison opinions can be expressed in a succinct format—the comparative relation.

Definition. A *comparative relation* is a formal expression of customers' comparison opinions, which can capture the customers' sentiment polarities on the competitive products about special attributes. A comparative relation can be expressed as a 4-tuple:

$$R\ (P1, P2, A, S)$$

where *P1* and *P2* are the two product names, *A* is the attribute name, and *S* is the sentimental phrase. For convenience, here we call all product names *P1* and *P2*, the attribute name *A*, and the sentimental phrase *S* as entity.

R is the direction of the comparative relation. Here, there are three optional directions: *Better* (>), *Worse* (<), and *Same* (=). *Better (>)*/*Worse (<)* means the product *P1* is better/worse than the product *P2* on the attribute *A* with the sentiment *S*; *Same* (=) means *P1* and *P2* are similar on *A* with *S*. Here, we take *No_Comparison (~)* as a special kind of direction that means there is no comparative relation between the two products.

Given an opinion sentence *O* consisting of two product names *P1* and *P2*, an attribute name *A*, and a sentimental phrase *S*, the comparative relation extraction task is 1) to detect if there is a comparative relation *R(P1, P2, A, S)*; and 2) if so, recognize the direction of *R*.

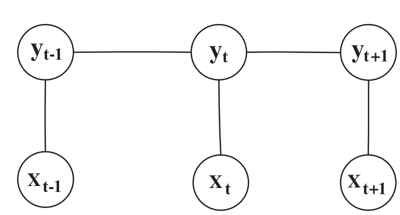

Fig. 1. Graphical representation of linear-chain CRF.

3.2. General process

The general process of building a comparative relation map from customer opinions is as follows (Fig. 2):

Data collection. Various raw customer reviews are collected from several sources: 1) online shopping sites, such as Amazon; 2) customer review sites, such as epinions; 3) blogs; 4) social network sites; and 5) emails in Customer Relation Management (CRM). For resources 1) and 2), the opinion data can be directly located and downloaded. For resources 3), 4), and 5), the techniques for identifying opinions can be used [43,45,46] (however, this is not the focus of this paper).

Feature extraction. Various linguistic features can be extracted from reviews for later use. Some preprocessing steps are carried out on the opinion data, including tokenization, sentence splitting, word stemming, syntactic tree parsing, dependency parsing, and so forth. In this study, we mainly consider some simple and effective features; the more complex linguistic features (such as semantic role) are not considered because user-generated reviews are always in informal expressions (even including some typos), and deep parsing is often inaccurate for this data. The linguistic features used include some basic linguistic features as well as more advanced ones:

1) Capitalization information: if the letters are capitalized. This is very useful for recognizing product names;
2) Word kind: whether the word is a number (such as "95"), or punctuation (such as ","). This is helpful in product name recognition;
3) Prefixes and suffixes: the prefixes and suffixes of words, such as "er" or "est". This feature is useful in recognizing sentiment phrases and comparative relations.
4) POS tags of words and phrase chunking: most product names and attribute names are nouns, and sentiment words are adjectives or adverbs. Also, the comparative adjective/adverb and superlative adjective/adverb are good indicators for the comparative relations.
5) Indicators of the comparative relations: we build a lexicon of comparative words, including the following:
"*in contrast to*", "*unlike*", "*compare with*", "*compare to*", "*beat*", "*win*", "*exceed*", "*outperform*", "*prefer*", "*than*", "*as*", "*same*", "*similar*", "*superior to*", "*improvement over*", "*better*", "*worse*", "*best*", "*worst*", "*more*", "*most*", "*less*", "*least*", and so forth.
6) Syntactic paths: the comparative relations are usually expressed with certain syntactic patterns, which can be reflected using the syntactic paths from syntactic trees. Thus, the syntactic paths between entities are helpful in recognizing the comparative relations.
7) Grammatical roles: the various entities usually play different grammatical roles in the sentences with comparative opinions, and these grammatical roles are helpful for recognizing the directions of comparative relations. For example, product entities are usually the subjects or prepositional phrase types, attribute entities are the objects, and sentiment entities are predicates. The grammatical roles are derived from the dependency parsing.

For these linguistic features, the Natural Language Processing (NLP) utility tools, Gate [37] and the Stanford Parser [38], are used to automatically annotate with high accuracy.

Entity recognition. The three types of entities should be recognized as follows: product names, attribute names, and sentiment phrases. In the customer opinion data, some product names always occur in various abbreviations; for example, "*BlackBerry 8320*" is written as "*BB 8320*" and "*8320*". Usually the product names have some special naming rules in the mobile phone domain, which is our focus at present. Also, the attribute names are finite and fixed in this domain, so it is easy to recognize them. Since this is not the main focus of this paper, the simpler, lexicon-based method is used for recognizing the names. The lexicons for product and attribute are built by collecting the common mobile phone names and attributes. For the sentiment phrases, a number of available lexical resources [11,12,44] are directly used to recognize them, since these lexicons include most of the sentiment phrases and their polarities. Another problem is that many pronouns (such as *it*, *they*, *them*, *both*, etc.) in the opinion data refer to product names or attribute names, and they should be resolved. Here the simple "closest-first" method is adopted.

Comparative relation extraction. This is the key and most difficult step. As mentioned in the above section, the particular characteristics of comparative relation extraction—involvement of multiple entities and recognizing the directions—make this problem more

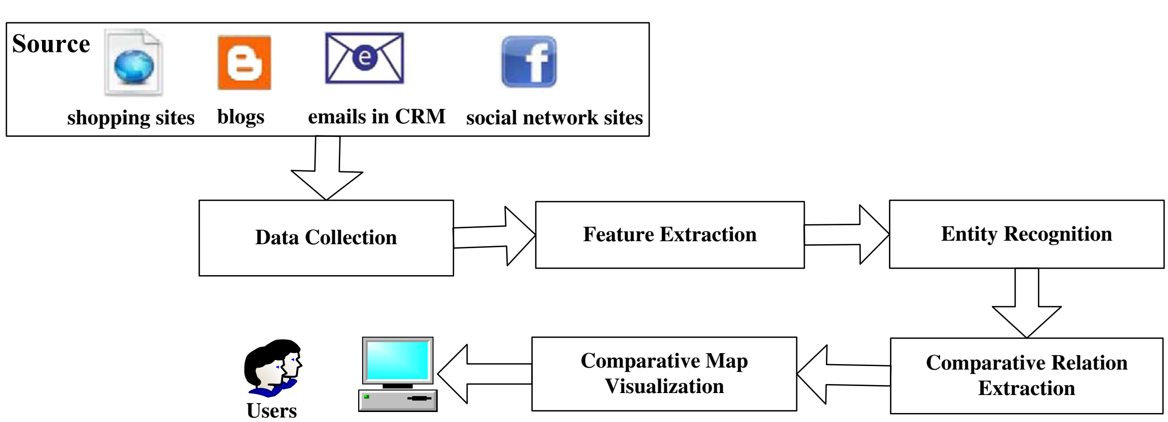

Fig. 2. General process of building a comparative relation map.

complicated than the existing relation extraction tasks. We will introduce a novel graphical model we propose for comparative relation extraction in Section 4.

Comparative map visualization. A number of postprocess steps need to be executed. These steps include 1) merging the different names of the same product and the same attribute—for example, the name of the mobile phone *Nokia N95* is sometimes written as "*N95*", or even "*95*", and they should be substituted with a consistent and formal name; 2) counting the number of the comparative relations between a pair of competitive products; and 3) summarizing and representing the comparative relations as comparative relation maps. These comparative relation maps can be shown visually and will be user-friendly from different views, according to the requirements of users, for supporting risk management and decision making.

4. Comparative relation extraction using a two-level CRF with unfixed interdependencies

4.1. Comparative relations

The following are several examples of the comparison opinions and their comparative relations.

Example 1: *Nokia N95 has a better camera than iPhone.*
> (Nokia N95, iPhone, camera, better)

Example 2: *Compared with Nokia N95, iPhone has a better camera.*
< (Nokia N95, iPhone, camera, better)

Example 3: *The Pearl and the Curve are both with high resolution camera.*
~ (Pearl, Curve, camera, high resolution)

Example 4: *The screen of iPhone is bigger than that of the curve, so I can read easily.*
> (iPhone, curve, screen, bigger)

Example 5: *The price of iPhone is much higher than that of the curve, so I can not afford it.*
< (iPhone, curve, price, higher)

Detecting the occurrence of comparative relations and recognizing their directions are more complicated problems than the existing relation extraction tasks. The existing relation extraction tasks mainly detect if a special relation exists between two entities or not, which is a typical binary-class classification problem. The comparative relation extraction needs not only to detect the occurrence but also to recognize the directions. This is a typical multi-class classification problem. To avoid executing two independent subtasks, we merge the detection of the occurrences and the recognition of the directions together, by taking *No_Comparison* (~) as a special kind of direction.

The above examples show that comparative relation extraction is a non-trivial task. For example, Examples 1 and 2 have the same entities, only with different arrangements, but they express two relations with different directions. Sometimes even the sentences with similar entities arrangements can express the relations of different directions, e.g., Examples 4 and 5. These examples show that the comparative relation extraction depends on both entity features (e.g., entity types and arrangements) and word features (e.g., keywords, sentiment words, and POS tags). One comparative relation contains four entities, which usually are widely distributed over the whole sentence, so these features are usually in the long range. In order to better extract the comparative relations, the tool resorted to should have a powerful capability of modeling the long-range dependencies.

In addition, in many customer opinions, one sentence contains multiple comparative relations, and these relations may be interdependent. Here are several examples,

Example 6: *N95 has better reception than Motorola RAZR2 V8 and Blackberry Bold 9000.*
r1: *> (N95, Motorola RAZR2 V8, reception, better)*
r2: *> (N95, Blackberry Bold 9000, reception, better)* (Fig. 3)

Example 7: *iPhone beats the curve in both function and looks.*
r1: *> (iPhone, Curve, function, beat)*
r2: *> (iPhone, Curve, looks, beat)* (Fig. 4)

Example 8: *The iPhone has better looks, but a much higher price than the BB Curve.*
r1: *> (iPhone, BB Curve, looks, better)*
r2: *< (iPhone, BB Curve, price, higher)* (Fig. 5)

Example 9: *Compared with the v3, this V8 has a bigger body, and it has a much worse keyboard than Nokia E71.*
r1: *> (V3, V8, body, bigger)*
r2: *< (V8, Nokia E71, keyboard, worse)* (Fig. 6)

For multiple comparative relations in one sentence, the directions of the relations may influence each other in some situations. In Example 6, for the relations r1 and r2, the entities "*Motorola RAZR2 V8*" and "*Blackberry Bold 9000*" are connected by the word "*and*" and play similar roles. Obviously, the directions of r1 and r2 tend to be the same. That is, if the direction of r1 is known, the direction of r2 can be inferred easily. It is a similar case for Example 7. In Example 8, the entities "*better looks*" and "*higher price*" are connected by the word "*but*", so the directions of r1 and r2 tend to be opposite. These examples indicate that there are interdependencies among the relations in one sentence, and these interdependencies are very useful in recognizing the directions of the comparative relations: If the direction of one relation is known, the directions of the other relations interdependent with that relation can be inferred easily by utilizing the interdependencies. However, the interdependencies among relations do not hold for all situations. In Example 9, the relations r1 and r2 are not interdependent. That is, the interdependencies are unfixed and conditional on the input. For better extracting of the comparative relations, the unfixed interdependencies among relations should be exploited. Especially when the training examples are limited and the entity and word features alone are not enough for recognizing some comparative relations, these unfixed interdependencies will have a major effect on recognizing them. The existing relation extraction methods [18,36] always assume that one sentence contained at most one relation, without considering the interdependencies among relations.

4.2. A graphical model for comparative relation extraction

The graphical model, CRF [25,39], is a powerful tool to model the complicated and long-range dependencies in an intuitive way. Unfortunately, current CRF models are one level or can only model fixed dependencies. In order to overcome the deficiencies of these models in comparative relation extraction, we propose a two-level CRF with unfixed interdependencies. This model can model the dependencies between relations and entities as well as between relations and words, for capturing the features from entity and word levels. Furthermore, this model is flexible enough to model the dependencies between entities and words. Therefore, it can concurrently recognize comparative relations and entities, which makes the

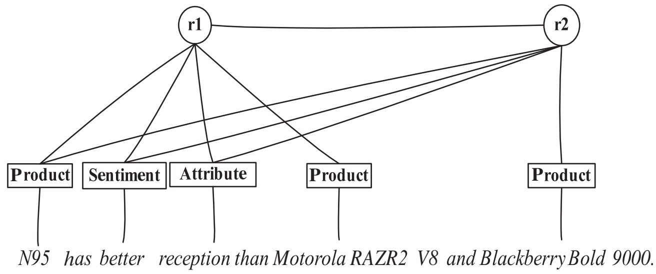

Fig. 3. The dependency graph for Example 6.

graphical model a two-level CRF. In this model, the unfixed interdependencies between relations are modeled by the edges conditional on the inputs. The graphical model is shown in Fig. 7.

In this graph, r_t is the candidate relation at the position t, r_{t-v} is the candidate relation at the position $t-v$ (the v preceding position of t) in one sentence, $e_n \ldots e_{n+3}$ and $e_l \ldots e_{l+3}$ are the entities at the positions of $n \ldots n+3$ and $l \ldots l+3$, and $w_k \ldots w_{k+3}$, w_i, and $w_m \ldots w_{m+3}$ are the words at the positions of $k \ldots k+3$ and $m \ldots m+3$. The edges linking the relation nodes with the entity nodes represent the dependencies between the relations and entities; the edges linking the relation nodes with the word nodes represent the dependencies between the relations and words (although the relations can depend on any words, we only show partial edges for clarity), and the dashed-line edges represent the unfixed interdependencies between the relations, which are conditional on the inputs (indicated by the black diamond). The advantage of this model is that it is an input-specific model structure. That is, for different inputs, the independencies between relations sometimes exist, and sometimes do not. For the inputs, if the entities of the relation r_t are connected to the entities of the relation r_{t-v}, by some special conjunctions (such as "*and*", "*or*", "*but*", "*while*", etc.), and these entities play similar roles, the interdependencies among the relations exist, like in Examples 6, 7 and 8. Otherwise, the interdependencies do not hold true, like in Example 9.

Let r_t ($t=1,\ldots T$, T is the number of candidate relations) be the direction of the candidate comparative relation in one sentence, $\mathbf{e}$ be the entities in this sentence, and $\mathbf{w}$ be the words of this sentence. For this two-level CRF with unfixed interdependencies, the probability of the directions of all candidate relations $\mathbf{r}=r_1 \ldots r_T$ is modeled as

$$P(\mathbf{r}|\mathbf{e},\mathbf{w}) = \frac{1}{Z(\mathbf{e},\mathbf{w})} \prod_{t=1}^{T} \Psi_t(r_t,\mathbf{e},\mathbf{w}) \prod_{t=1}^{T} \prod_{1 \le v < t} \Phi_t(r_t, r_{t-v}, \mathbf{e}, \mathbf{w}) \tag{1}$$

where Ψ_t is the factor over the edges between the relations, the entities, and the words; Φ_t is the factor over the edges between the relations (the dashed-line edges); and $Z(\mathbf{e},\mathbf{w})$ is the normalization function.

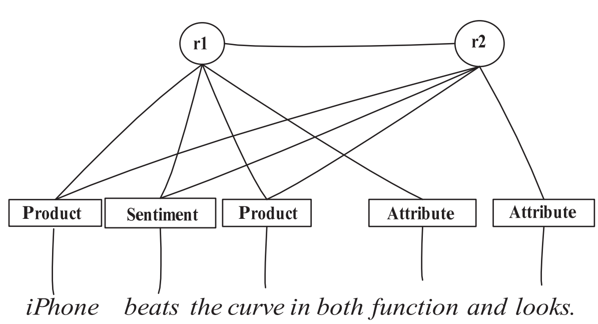

Fig. 4. The dependency graph for Example 7.

The factor $\Psi_t(r_t,\mathbf{e},\mathbf{w})$ is defined as

$$\Psi_t(r_t,\mathbf{e},\mathbf{w}) = \exp\left\{ \sum_k \lambda_k^e f_k^e(r_t,\mathbf{e}) + \sum_k \lambda_k^w f_k^w(r_t,\mathbf{w}) \right\}. \tag{2}$$

The parameters λ_k^e and λ_k^w (k takes integer values) are the weights for entity features and word features, respectively. The functions $f_k^e(r_t,\mathbf{e})$, are the feature functions about entity features, and the functions $f_k^w(r_t,\mathbf{w})$, are the feature functions about word features.

The factor $\Phi_t(r_t,r_{t-v},\mathbf{e},\mathbf{w})$ is defined as

$$\Phi_t(r_t,r_{t-v},\mathbf{e},\mathbf{w}) = \exp\left\{ u(r_t,r_{t-v},\mathbf{e},\mathbf{w}) \left(\sum_k \lambda_k^r f_k^r(r_t,r_{t-v}) \right) \right\} \tag{3}$$

where λ_k^r (k takes integer values) are the weight parameters. The functions $f_k^r(r_t,r_{t-v})$ are the feature functions about the two relations. $u(r_t,r_{t-v},\mathbf{e},\mathbf{w})$ is the indicator function to indicate whether the interdependency between r_t and r_{t-v} holds true or not. As mentioned earlier, if the entities of r_t are connected with the entities of r_{t-v}, using some special conjunctions, and these entities play the same roles in the inputs, the value of the indicator function is 1; otherwise it is 0. Although in the formula (Eq. 1), v can take multiple values from 1 to $t-1$, the indicator function makes the connections between the relation nodes very limited, which ensures that the inference of the graphic model is feasible.

For the two-level CRF with unfixed interdependencies, the belief propagation algorithm [39] can be adopted as the approximate inference method. To estimate the parameters λ_k^e, λ_k^w and λ_k^r, the maximum conditional log likelihood [39] can be used. Regularization can be adopted to avoid overfitting.

4.3. Illustrative examples

Here, we use two special examples to illustrate the above formulas. Assume there is only one feature function each for three groups of feature functions, $f_k^e(r_t,\mathbf{e})$, $f_k^w(r_t,\mathbf{w})$, and $f_k^r(r_t,r_{t-v})$, and the three feature functions are defined as:

$$f_{k1}^e(r_t,\mathbf{e}) = 1_{\{r_t='>'\}} 1_{\{e_1='P'\}} 1_{\{e_2='S'\}} 1_{\{e_3='A'\}} 1_{\{e_4='P'\}} = \begin{Bmatrix} 1 & r_t \text{ is } >\ \text{first entity is P, second entity is S, third entity is A, and fourth entity is P} \\ 0 & \text{Otherwise} \end{Bmatrix} \tag{4}$$

here, the indicator function $1_{\{x=x'\}}$, takes the value 1 only $x=x'$.

$$f_{k1}^w(r_t,w) = 1_{\{r_t='<'\}} 1_{\{\text{sentiment word}='\textit{worse}'\}} = \begin{Bmatrix} 1 & r_t \text{ is } < \text{ and the sentiment word is "worse"} \\ 0 & \text{Otherwise} \end{Bmatrix} \tag{5}$$

$$f_{k1}^r(r_t,r_{t-v}) = 1_{\{r_t='>'\}} 1_{\{r_{t-v}='>'\}} = \begin{Bmatrix} 1 & r_t \text{ and } r_{t-v} \text{ are both} > \\ 0 & \text{Otherwise} \end{Bmatrix} \tag{6}$$

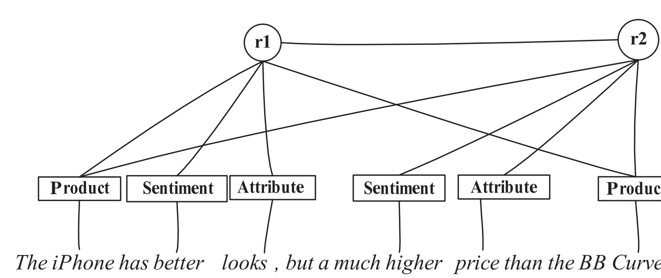

Fig. 5. The dependency graph for Example 8.

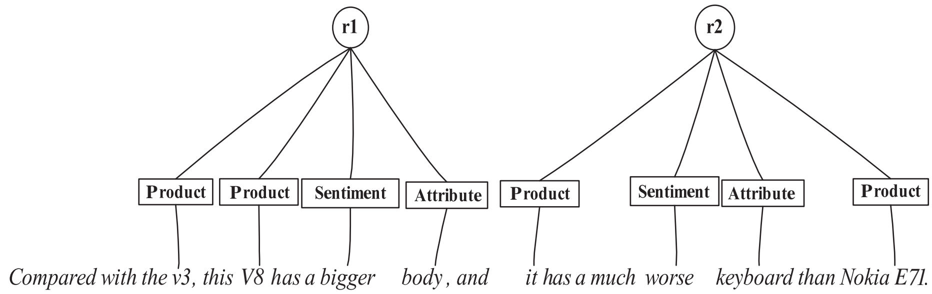

Fig. 6. The dependency graph for Example 9.

λ^e_{k1}, λ^w_{k1} and λ^r_{k1} are the weights for the three features, and their values are respectively 0.2, 0.1 and 0.25 after the parameter estimation.

For Example 6, if the directions of the two relations both take ">", because the entities of the two relations occur in the sentence in the order Product, Sentiment, Attribute and Product, the conditions of all indicator functions in Eq. (4) are satisfied. So f^e_{k1} is 1 for the two relations. For f^w_{k1}, the two indicator functions in Eq. (5) cannot be satisfied, so f^w_{k1} is 0 for the two relations. It is obvious that the indicator functions in Eq. (6) are satisfied, so f^r_{k1} is 1. Also, r_2 and r_1 satisfy the interdependency condition, so $u(r_2, r_1, \mathbf{e}, \mathbf{w}) = 1$. Thus, the probability of the two directions both being ">" is

$$P(>>|\mathbf{e},\mathbf{w}) = \frac{1}{Z(\mathbf{e},\mathbf{w})}\underbrace{\exp\{0.2\times1+0.1\times0\}}_{\Psi_1}\underbrace{\exp\{0.2\times1+0.1\times0\}}_{\Psi_2}\underbrace{\exp\{1\times(0.25\times1)\}}_{\Phi_2}$$

If the direction of the first relation takes ">", and the direction of the second takes "<", f^e_{k1} is 1 for the first relation and is 0 for the second relation, since the first indicator function in Eq. (4) cannot be satisfied; f^w_{k1} is still 0 for the two relations, and f^r_{k1} is 0. So the probability of this direction option is

$$P(><|\mathbf{e},\mathbf{w}) = \frac{1}{Z(\mathbf{e},\mathbf{w})}\underbrace{\exp\{0.2\times1+0.1\times0\}}_{\Psi_1}\underbrace{\exp\{0.2\times0+0.1\times0\}}_{\Psi_2}\underbrace{\exp\{1\times(0.25\times0)\}}_{\Phi_2}$$

It is obvious that $P(>>|\mathbf{e},\mathbf{w})$ is greater than $P(><|\mathbf{e},\mathbf{w})$. That is, it is more possible that the two directions are both ">". (Here, $Z(\mathbf{e},\mathbf{w})$ is a constant to make sure the probability value is between 0 and 1.)

For Example 9, if the direction of the first relation takes ">", and that of the second takes "<", f^e_{k1} is 0 for the two relations, and f^w_{k1} is 0 for the first relation and 1 for the second relation. r_2 and r_1 do not satisfy the interdependency condition, so $u(r_2, r_1, \mathbf{e}, \mathbf{w}) = 0$. Thus, the probability of this direction option is

$$P(><|\mathbf{e},\mathbf{w}) = \frac{1}{Z(\mathbf{e},\mathbf{w})}\underbrace{\exp\{0.2\times0+0.1\times0\}}_{\Psi_1}\underbrace{\exp\{0.2\times0+0.1\times1\}}_{\Psi_2}\underbrace{\exp\{0\times(0.25\times0)\}}_{\Phi_2}$$

5. Experimental evaluation

We conducted experiments to evaluate the performances of the proposed model for comparative relation extraction. Furthermore, a case/tutorial is used to show the usefulness of the comparative relation maps for risk management and decision support.

5.1. Data collection and annotation

The raw customer opinions data were collected from the Amazon online shopping site. Our test bed included 1347 customer reviews of 33 types of mobile phones. Three domain experts in mobile phones were employed to manually annotate these opinion data. First, they extracted the segments containing two different product names, and these segments usually included all potential comparative relations. Second, for the extracted segments, they annotated the product names, attribute names, and sentiment phrases. Third, they detected whether there were comparative relations among different products in one sentence. If so, the directions of the comparative relations were recognized; otherwise, they were labeled as *No_Comparison (~)*. The final annotating results had to reach agreement by at least two persons. In summary, the dataset consists of 1608 product names, 783 attribute names, and 933 sentiment phrases. These entities comprise 1098 comparative relations: 525 *Better* (>) relations, 345 *Worse* (<) relations, 78 *Same* (=) relations, and 150 *No_Comparison* (~) relations. Of these, there are 549 single-relation sentences, 150 two-relation sentences (for these sentences, each includes two relations), and 39 three-relation sentences. The summary statistics of the dataset are shown in the following table (Table 1).

5.2. Evaluation settings and criteria

5.2.1. Evaluation setting

Evaluation 1. In order to evaluate the performance of our proposed model for comparative relation extraction, we compared it with a multi-class classification technique as a benchmark: the multi-class

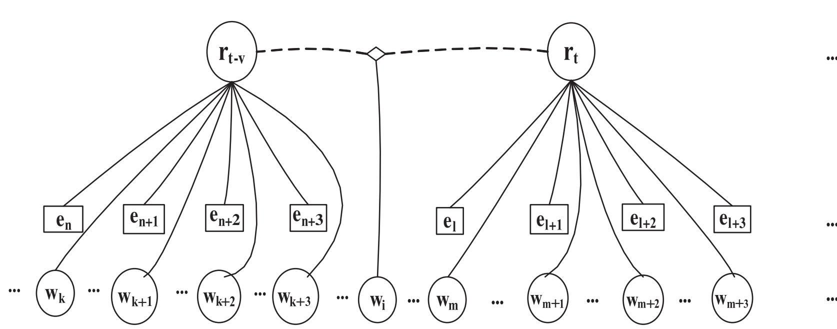

Fig. 7. Graphical representation of the two-level CRF with unfixed interdependencies.

Table 1
Summary of the dataset.

	Product	Attribute	Sentiment	Total	
Entity type	1608	525	933	2526	
	"Better"	"Worse"	"Same"	"No_Comparison"	Total
Relation type	525	345	78	150	1098
	Single-relation sentence	Two-relation sentence	Three-relation sentence	Others	
Relation dependency	549	150	39	30	

Table 2
Performances of three methods.

Methods	Accuracy (%)	Direction	Precision (%)	Recall (%)	F-score (%)
Multi-class SVM	61.38	>	61.96	93.49	74.26
		<	54.44	39.87	45.43
		=	0	0	0
		~	66.67	25	35.56
		Average	45.77	39.59	38.81
CRF without interdependencies	60.04	>	79.16	76.59	77.35
		<	54.14	41.88	45.32
		=	23.33	16.67	19.39
		~	38.03	70.83	48.79
		Average	48.67	51.49	47.71
CRF with interdependencies	66.17	>	76.63	81.90	78.56
		<	59.72	48.39	51.77
		=	61.11	23.33	31.19
		~	54.29	83.33	65.20
		Average	62.94	59.24	56.68

SVM [41]. The multi-class SVM is a popular multi-class classification technique that has demonstrated good performance in many studies [16,30]. For this reason, we chose it as a benchmark to evaluate the proposed model. The "one-against-all" multi-class SVM was adopted. If a sentence contains multiple candidate relations, these potential relations with the corresponding entity and word features will be treated as different instances and be separately fed into the SVM classifier. Here is an example: "*The iPhone has better looks, but a much higher price than the BB Curve.*". In this sentence, there exist two relations: > *(iPhone, BB Curve, looks, better)* and <*(iPhone, BB Curve, price, higher)*. For the two relations, two instances are built for the SVM classifier with the following feature vectors:

class_label:> first_entity_type:P; second_entity_type:S; third_entity_type:A; fourth_- entity_type:P; sentiment_word: better; …(other features).

class_label:< first_entity_type:P; second_entity_type:S; third_entity_type:A; fourth_entity_type:P; sentiment_word: higher; …(other features).

Here, the first and second entities refer to the order that the entities occur in the sentence. These two instances are treated as independent instances to be fed into the classifier. Also, the proposed model was compared with a two-level CRF without considering the interdependencies among relations, for evaluating the influence of unfixed interdependencies. In this evaluation, the used linguistic features include the following: 1) indicator words, 2) words for entities, 3) entity types, 4) phase types of entities, 5) syntactic paths between adjacent entities, and 6) grammatical roles of entities.

Evaluation 2. This evaluation was made to compare the proposed model with Jindal and Liu's method (marked as J&L's method) in [18] for extracting user opinions. Their work mainly divides the comparative relations into four categories: "*non-comparison*", "*non-equal gradable*", "*same*", and "*superlative*", but does not differentiate the directions of "*non-equal gradable*" relations. Because in this dataset, the "*superlative*" relations are very few, this evaluation mainly compared the performances of different methods in recognizing "*non-comparison*", "*non-equal gradable*", and "*same*" relations. Here, some basic linguistic features, indicator words and POS tags, were used, as in [18].

Evaluation 3. The effects of different linguistic features on extracting comparative relations also were evaluated. This evaluation mainly compared the performance of the proposed method when only using the basic linguistic features to its performance when using the basic and advanced linguistic features together. (The basic features include the following: 1) indicator words, 2) words for entities, and 3) entity types; the advanced features include 1) phase types of entities, 2) syntactic paths between adjacent entities, and 3) grammatical roles of entities.)

5.2.2. Evaluation criteria

The performance was measured using four metrics: accuracy, precision, recall, and F-score. Accuracy measures the overall correctness of the classification.

$$\text{Accuracy} = \frac{\#\text{ of correctly extracted relations}}{\text{total }\#\text{ of relations}}$$

Precision, recall, and F-score are used to evaluate the correctness of extracted relations in every kind of direction. For a kind of direction, the precision and recall are defined as

$$\text{Precision} = \frac{\#\text{ of correctly extracted relations of the direction}}{\text{total }\#\text{ of extracted relations of the direction}}$$

$$\text{Recall} = \frac{\#\text{ of correctly extracted relations of the direction}}{\text{total }\#\text{ of relations of the direction}}.$$

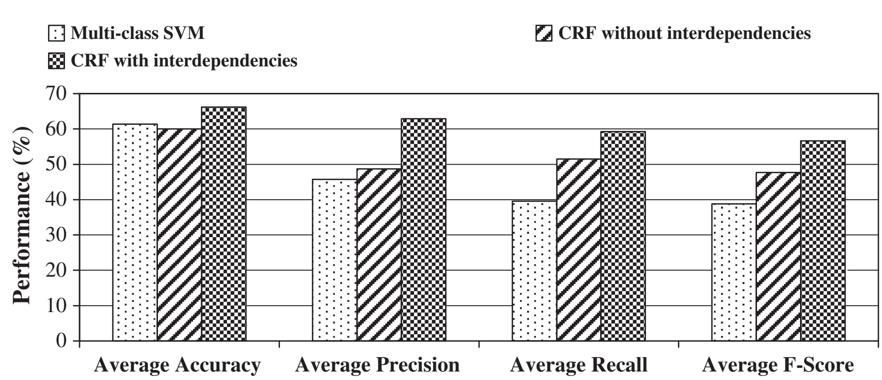

Fig. 8. Performance comparison of three methods.

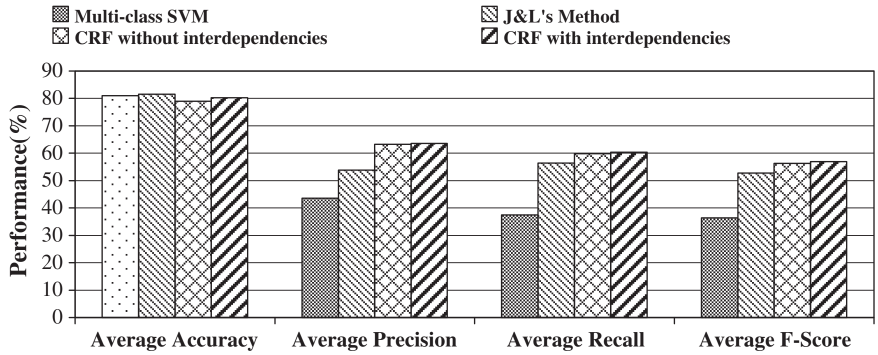

Fig. 9. Performance comparison with J&L's method.

F-score is the harmonic mean of precision and recall,

$$\mathrm{F-score} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}.$$

Following the traditional method [32], the standard 5-fold cross-validation was used, in which the dataset is randomly partitioned into 5 subsets; 4 of them are used as training data, and the final one is used as the validation data. The average 5-fold cross-validation metrics were used as the reported results.

The SVM multi-class [20] and CRF toolbox [29] software tools were used in our experiments. Previous research [19] has shown that a linear kernel can achieve similar performance to non-linear kernels (e.g., RBF) for text classification using SVM. Therefore, in our experiments we chose the linear kernel and set its parameter c as 1; for the CRF, the regulation with the value 0.5 was used to avoid overfitting.

5.3. Experiment results and discussions

5.3.1. Relation direction recognition

Table 2 and Fig. 8 show the accuracies, precisions, recalls, and F-scores of the three methods for recognizing the relation directions.

As the table and the figure show, the performances of the two-level CRF with unfixed interdependencies are clearly better than those of the multi-class SVM and CRF without interdependencies. The two-level CRF with unfixed interdependencies achieves 66.17% average accuracy, which is an improvement of 7.80% and 10.21% when compared, respectively, to the multi-class SVM and CRF without interdependencies. In addition, the CRF with unfixed interdependencies increases the average precision by 22.67%, the average recall by 15.05%, and the average F-score by 18.8%.

Let us look at one example from the test data:

Comparing the Nokia E71 with my iPhone 3G, the E71 has a better reception, but smaller screen size.

In this example, there are two candidate relations:

r1: >(Nokia E71, iPhone 3G, reception, better)
r2: <(Nokia E71, iPhone 3G, smaller, screen size)

The two-level CRF without interdependencies and multi-class SVM recognized r1 and r2 both as ">". But the two-level CRF with unfixed interdependencies correctly recognized them as ">" and "<", respectively. And the weight parameter for the feature of the relations ">" and "<" was positive. This indicates that there is a positive correlation between ">" and "<", and ">" can reinforce the possibility of recognizing "<". Thus, the CRF with unfixed interdependencies can correctly recognize "<". These experiment results show that the two-level CRF with unfixed interdependencies is very effective in improving the performance of identifying the comparative relations and recognizing the directions.

5.3.2. Comparison with the prior method

The performance comparison between the proposed method and J&L's method [18], in recognizing "*non-comparison*", "*non-equal gradable*", and "*same*" relations, is shown in Fig. 9 and Table 3.

The figure indicates that the proposed method results in better average precision, recall, and F-score compared to J&L's method. The table indicates that J&L's method has higher average accuracy and better recall in recognizing "*unequal gradable*" relations, but its

Table 3
Performances of the proposed method and J&L's method.

Methods	Accuracy (%)	Direction	Precision (%)	Recall (%)	F-score (%)
Multi-class SVM	81.04	Non-comparison	50	12.5	20
		Non-equal gradable	80.76	100	89.29
		Same	0	0	0
		Average	43.59	37.5	36.43
J&L's method	81.56	Non-comparison	49.34	68.65	56.71
		Non-equal gradable	86.75	91.94	88.83
		Same	25.56	8.67	12.86
		Average	53.88	56.42	52.80
CRF without interdependencies	78.94	Non-comparison	48.08	80	58.85
		Non-equal gradable	91.67	84.38	86.97
		Same	50	15	23.08
		Average	63.25	59.79	56.30
CRF with interdependencies	80.22	Non-comparison	49	80	59.71
		Non-equal gradable	91.67	85.94	87.98
		Same	50	15	23.08
		Average	63.56	60.31	56.92

Table 4
Performances when using different linguistic features.

Methods	Accuracy (%)	Direction	Precision (%)	Recall (%)	F-score (%)
Basic features	62.76	>	75.96	61.36	67.43
		<	45.00	53.17	47.55
		=	27.00	31.90	28.53
		~	62.00	76.67	67.52
		Average	52.51	55.78	52.76
Basic + advanced features	66.17	>	76.63	81.9	78.56
		<	59.72	48.39	51.77
		=	61.11	23.33	31.19
		~	54.29	83.33	65.20
		Average	62.94	59.24	56.68

performance in extracting "*same*" relations is poor. One possible reason may be that J&L's method adopts the SVM classifier to distinguish "*non-comparison*" and "*same*" relations in one step. As the table indicates, the SVM method cannot handle the unbalanced problem very well: opinion data contain far more "*unequal gradable*" relations than "*same*" relations. Also in this evaluation, the CRF without interdependencies results in a similar performance to the CRF with interdependencies. The reason may be that the simple word and POS tag features are enough to distinguish "*unequal gradable*" and "*same*" relations, so the interdependencies of relations do not play an obvious role in this aspect. However, in recognizing the relation directions, these interdependencies show important effects.

5.3.3. Effects of linguistic features

The performances when using different linguistic features are shown in Table 4 and Fig. 10.

The table and figure indicate that, when only the basic linguistic features are used, the method achieves acceptable performances. For example, the average accuracy is 62.76%, and the average precision, recall, and F-score are more than 50%. After integrating the advanced linguistic features, the performances increase substantially. The average precision reaches 62.94% with an increase of 19.86%, and the average recall and F-score increase by 6.2% and 7.43%, respectively. This indicates that the advanced features also play important roles in extracting comparative relations.

5.4. A comparative relation map scenario

After the comparative relations are extracted, the comparative relation maps can be built for decision support. Here, such a scenario/case is used to demonstrate the usefulness of the comparative relation map in enterprise risk management: Nokia has published the smart phone E71, while competitive producers have also released similar products. The manager at Nokia wants to know the customers' sentiments on the E71, as compared with their sentiments on several competitive products, in order to design a market strategy (such as price reduction, targeting particular customers) and develop the next version with stronger points. Extracting the comparative opinions with the proposed method can aid the manager in achieving this object. The extracted comparative relations from user opinions can be summarized and visualized as comparative relation maps in an intuitive and user-friendly way. The following figure shows a screen shot of the comparative relation map for this scenario/case (Fig. 11).

In the comparative relation map, the red lines represent the numbers of ">" comparative relations for the product that is of interest to managers, while the blue lines represent the numbers of ">" comparative relations for competitive products. The manager can choose to only show the specific competitive products and attributes with which he is concerned. The original customer reviews can be easily navigated for the manager's reference. In the comparative relation map, the manager can intuitively learn the relative strengths and weaknesses of their product, which are graded directly by the customers. As the response, the manager might make some adjustment on a market strategy or a new version of the product. This scenario/case shows that the comparative relation map is an intuitive and effective way in aiding the enterprise managers in discovering operation risks and supporting appropriate decisions.

6. Conclusions and future work

In this paper, we designed a novel method to extract comparative relations from customer opinion data, to build comparative relation maps for aiding enterprise managers in identifying the potential operation risks and supporting strategy decisions. The two-level CRF model with unfixed interdependencies can better extract the comparative relations, by utilizing the complicated dependencies between relations, entities and words, and the unfixed interdependencies among relations. The empirical evaluation demonstrated the effectiveness of this model. The comparative relation map is potentially a very effective tool to support enterprise risk management and decision making. The contributions of this paper include the following: 1) To the best of our knowledge, this is the first work on using comparison opinion as information sources in CI for enterprise risk management; 2) the proposed graphical model can achieve better performance for relation extraction by modeling the unfixed interdependencies among relations, which is not covered by the existing methods; and 3) the empirical evaluation shows that the performance of the comparative relation extraction is quite promising, and it implies the feasibility of mining the comparison opinions for CI.

In the future, we plan to conduct an empirical evaluation of the proposed model on a larger scale with other product types. We also plan to extend the model to jointly recognize the comparative relations and entities so as to reduce the errors accumulated in the pipeline process. In addition, the comparative relation map will be

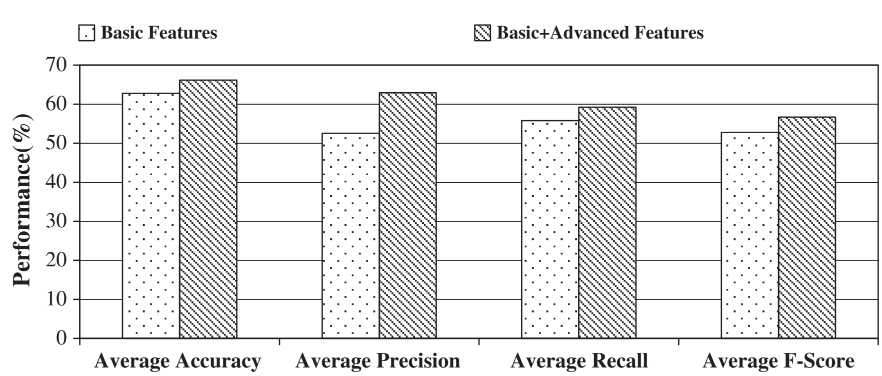

Fig. 10. Performance comparison when using different linguistic features.

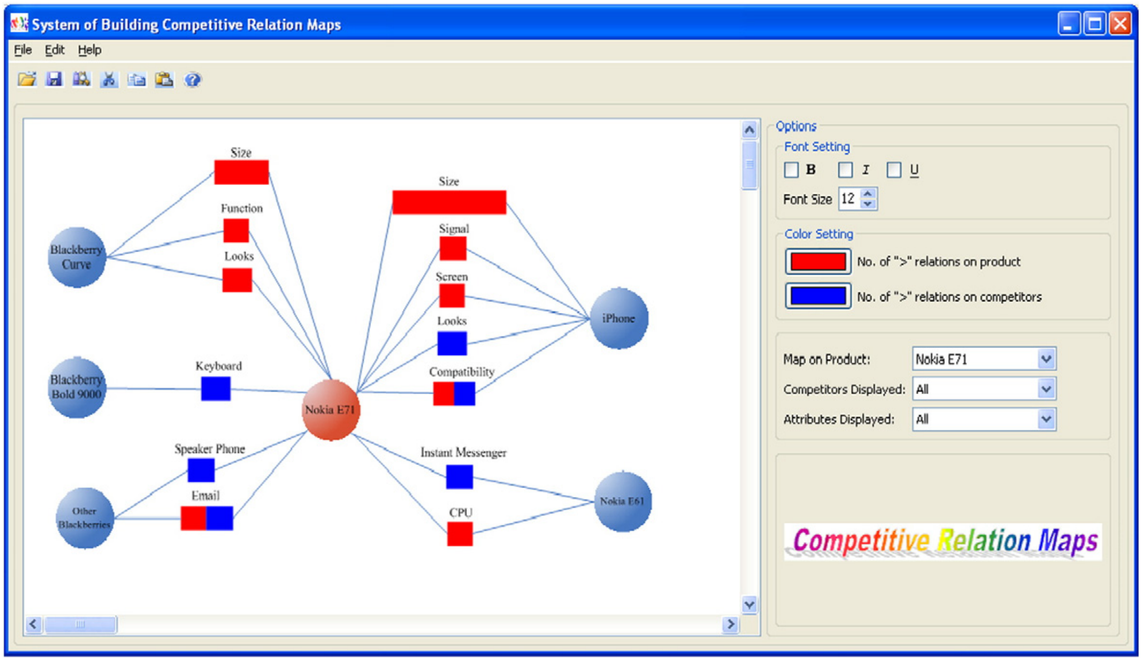

Fig. 11. Screen shot of a comparative relation map.

aligned with the product market shares, for facilitating the analysis of customer opinions' influence on the product sales and better support of enterprise decisions. Managers from the industry will be invited to use and evaluate the system.

Acknowledgements

This research was directly supported by a GRF grant (CityU 147407), of the RGC, Hong Kong SAR government and an SRG grant (7002425) of the City University of Hong Kong. The authors are also grateful to the referees for their helpful comments and valuable suggestions for improving the earlier version of the paper.

References

[1] A. Abbasi, H. Chen, A. Salem, Sentiment analysis in multiple languages: feature selection for opinion classification in Web forums, ACM Transactions on Information Systems 26 (3) (2008).
[2] S. Argamon, C. Whitelaw, P. Chase, S. Raj Hota, N. Garg, S. Levitan, Stylistic text classification using functional lexical features, Journal of the American Society for Information Science and Technology 58 (6) (2007).
[3] S. Bethard, H. Yu, A. Thornton, V. Hatzivassiloglou, D. Jurafsky, Automatic extraction of opinion propositions and their holders, in: Y. Qu, J. Shanahan, J. Wiebe (Eds.), Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications, Stanford University, USA, 2004, pp. 22–29.
[4] R.C. Bunescu, R.J. Mooney, A shortest path dependency kernel for relation extraction, in: R.J. Mooney (Ed.), Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, Vancouver, British Columbia, Canada, 2005, pp. 724–731.
[5] R. Bunescu, R. Mooney, Subsequence kernels for relation extraction, in: Y. Weiss, B. Schlkopf, J. Platt (Eds.), Advances in Neural Information Processing Systems, 18, The MIT Press, 2006.
[6] M. Chau, J. Xu, Mining communities and their relationships in blogs: a study of online hate groups, International Journal of Human–Computer Studies 65 (1) (2007).
[7] H. Chen, Intelligence and security informatics: information systems perspective, Decision Support Systems 41 (3) (2006).
[8] A. Culotta, J. Sorensen, Dependency tree kernels for relation extraction, in: D. Scott (Ed.), Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Barcelona Spain, 2004, p. 423.
[9] A. Divoli, T.K. Attwood, BioIE: extracting informative sentences from the biomedical literature, Bioinformatics 21 (2005).
[10] D. Edwards, Introduction to Graphical Modelling, 2nd ed.Springer, Verlag, 2000.
[11] A. Esuli, F. Sebastiani, SENTIWORDNET: a publicly available lexical resource for opinion mining, in: N. Calzolari (Ed.), Proceedings of the 5th Conference on Language Resources and Evaluation, Genoa, Italy, 2006, pp. 1–6.
[12] Z. Fei, extended SentimentWord, http://www.jkx.fudan.edu.cn/mcwil/~zcfei/sent2009.
[13] B.A. Hagedorn, M. Ciaramita, J. Atserias, World knowledge in broad-coverage information filtering, in: W. Kraaij, A.P. Vries (Eds.), Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, The Netherlands, Amsterdam, 2007, pp. 801–802.
[14] J. Hakenberg, C. Plake, U. Leser, H. Kirsch, LLL'05 challenge: genic interaction extraction—identification of language patterns based on alignment and finite state automata, in: S. Dzeroski (Ed.), Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 2005, pp. 38–45.
[15] V. Hatzivassiloglou, J.M. Wiebe, Effects of adjective orientation and gradability on sentence subjectivity, in: M. Kay (Ed.), Proceedings of the 18th Conference on Computational Linguistics, Saarbrücken, Germany, 2000, pp. 299–305.
[16] C.W. Hsu, C.J. Lin, A comparison of methods for multiclass support vector machines, IEEE Transactions on Neural Networks 13 (2) (2002).
[17] M. Hu, B. Liu, Mining and summarizing customer reviews, in: W. Kim, R. Kohavi (Eds.), Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 2004, pp. 168–177.
[18] N. Jindal, B. Liu, Mining comparative sentences and relations, in: K.D. Forbus (Ed.), Proceedings of 21st National Conference on Artificial Intelligence, Boston, Massachusetts, USA, 2006, pp. 1–6.
[19] T. Joachims, Text categorization with support vector machines: learning with many relevant features, in: C. Nedellec, C. Rouveirol (Eds.), Proceedings of the 10th European Conference on Machine Learning, Chemnitz, Germany, 1998, pp. 137–142.
[20] T. Joachims, SVM-multiclass: Multi-Class Support Vector Machine, http://svmlight.joachims.org2009.
[21] N. Kambhatla, Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations, in: J.S. Chang, Y.C. Chang (Eds.), Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions, Barcelona Spain, 2004, pp. 178–181.
[22] Y. Kim, Y. Jung, S. Myaeng, Identifying opinion holders in opinion text from online newspapers, in: T.Y. Lin, R. Yager, J. Mendel (Eds.), Proceedings of the 2007 IEEE International Conference on Granular Computing, San Jose, California, USA, 2007, pp. 699–702.
[23] L. Kong, Y. Fu, X. Zhou, Q. Liu, Z. Cui, Study on Competitive Intelligence system based on Web, in: Y. Fu (Ed.), Proceedings of the Workshop on Intelligent Information Technology Application, Zhang Jiajia, China, 2007, pp. 339–342.
[24] T. Kudo, Y. Matsumoto, A boosting algorithm for classification of semi-structured text, in: D. Lin, D. Wu (Eds.), Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, 2004, pp. 1–8.
[25] J.D. Lafferty, A. McCallum, F.C.N. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: C.E. Brodley, A.P. Danyluk (Eds.), Proceedings of the Eighteenth International Conference on Machine Learning, San Francisco, CA, USA, 2001, pp. 282–289.

[26] J. Li, Z. Zhang, X. Li, H. Chen, Kernel-based learning for biomedical relation extraction, Journal of the American Society for Information Science and Technology 59 (5) (2008).
[27] B. Liu, Web Data Mining—Exploring Hyperlinks, Contents and Usage Data, Springer, Verlag, 2006.
[28] B. Liu, M. Hu, J. Cheng, Opinion observer: analyzing and comparing opinions on the Web, in: A. Ellis, T. Hagino (Eds.), Proceedings of the 14th International World Wide Web Conference, Chiba, Japan, 2005, pp. 342–351.
[29] K. Murphy, Conditional Random Field (CRF) Toolbox for Matlab, http://www.cs.ubc.ca/~murphyk/Software/CRF/crf.html2004.
[30] M.N. Nguyen, J.C. Rajapakse, Multi-class support vector machines for protein secondary structure prediction, Genome Informatics 14 (2003).
[31] B. Pang, L. Lee, Opinion mining and sentiment analysis, in: J. Callan, F. Sebastiani (Eds.), Foundations and Trends in Information Retrieval, 2, Now Publishers, 2008.
[32] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? Sentiment classification using machine learning techniques, in: D. Yarowsky, D. Radev (Eds.), Proceedings of the Conference on Empirical Methods in Natural Language Processing, Philadelphia, PA, USA, 2002, pp. 79–86.
[33] M.E. Porter, Competitive Strategy: Techniques for Analyzing Industries and Competitors, Free Press, 1980.
[34] T.S. Raghu, H. Chen, Cyberinfrastructure for homeland security: advances in information sharing, data mining, and collaboration systems, Decision Support Systems 43 (4) (2007).
[35] E. Riloff, J. Wiebe, W. Phillips, Exploiting subjectivity classification to improve information extraction, in: M.V. Manuela, K. Subbarao (Eds.), Proceedings of the 20th National Conference on Artificial Intelligence, Pittsburgh, Pennsylvania, USA, 2005, pp. 1106–1111.
[36] B. Rosario, M.A. Hearst, Classifying semantic relations in bioscience texts, in: D. Scott (Ed.), Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 430–437.
[37] University of Sheffield, Natural Language Processing Research Group, GATE, A general architecture for text engineering, available at http://gate.ac.uk, (2009).
[38] Stanford Natural Language Processing Group, The Stanford Parser: a statistical parser, http://nlp.stanford.edu/software/lex-parser.shtml (2009).
[39] C. Sutton, A. McCallum, An introduction to conditional random fields for relational learning, in: L. Getoor, B. Taskar (Eds.), Introduction to Statistical Relational Learning, Ch.1, MIT Press, 2007.
[40] P.D. Turney, Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews, in: P. Isabelle (Ed.), Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 2001, pp. 417–424.
[41] J. Weston, C. Watkins, Multi-class Support Vector Machines, Technical Report, May 1998.
[42] J.M. Wiebe, R.F. Bruce, T.P. O'Hara, Development and use of a gold-standard data set for subjectivity classifications, in: R. Dale, K. Church (Eds.), Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, College Park, Maryland, USA, 1999, pp. 246–253.
[43] J. Wiebe, T. Wilson, R. Bruce, M. Bell, M. Martin, Learning subjective language, Computational Linguistics 30 (3) (2004).
[44] WordNet, http://wordnet.princeton.edu, (2009).
[45] J. Yi, T. Nasukawa, R. Bunescu, W. Niblack, Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques, in: J. Shavlik (Ed.), Proceedings of the Third IEEE International Conference on Data Mining, Melbourne, Florida, USA, 2003, pp. 427–434.
[46] J. Yi, W. Niblack, Sentiment mining in WebFountain, in: K. Aberer, M. Franklin, S. Nishio (Eds.), Proceedings of the 21st International Conference on Data Engineering, Tokyo, Japan, 2005, pp. 1073–1083.
[47] H. Yu, V. Hatzivassiloglou, Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences, in: M. Collins, M. Steedman (Eds.), Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, Sapporo, Japan, 2003, pp. 129–136.
[48] D. Zelenko, C. Aone, A. Richardella, Kernel methods for relation extraction, Journal of Machine Learning Research 3 (2003).

Kaiquan Xu is a Ph.D. student in the Department of Information Systems at City University of Hong Kong. His research interests include opinion mining, social network mining, information extraction, and machine learning. He has published articles in the Journal of Information Science and presented papers at conferences, such as ICIS, HICSS, AMCIS, and PACIS. He has also worked in the IT industry as a technical staff at Oracle China R&D Center.

Dr Stephen Liao is Associate Professor of Information Systems at the City University of Hong Kong. He earned a Ph.D. from University of Aix-Marseille III and the Institute of France Telecom in 1993. He has been working at the City University of Hong Kong since 1993, and his research has focused on use of IT in e-business systems. His articles have been published extensively in various academic journals such as Decision Support Systems, IEEE transactions, Communications of the ACM, Information Science, Computer Software, and so on. His current research interests include use of data mining techniques in mobile commerce applications and intelligent business systems, especially intelligent transportation systems.

Dr. Jiexun Li received his Bachelor's degree in Engineering in 2000 and Master's degree in Management in 2002 from Tsinghua University in Beijing, China. He received his Ph.D. degree in Management (with a major focus in Management Information Systems) from the University of Arizona in 2007. He is currently an Assistant Professor in the College of Information Science and Technology at Drexel University. His research interests are knowledge discovery (data/text/web mining), machine learning, network analysis, and their applications in various areas, including customer relationship management, bioinformatics, medical informatics, security, and so on.