future internet

Article

Sentiment Analysis Based Requirement Evolution Prediction

Lingling Zhao [1] and Anping Zhao [2,*]

[1] College of Computer and Information Science, Chongqing Normal University, Chongqing 401331, China; llzhao@cqnu.edu.cn
[2] Institute of Education Informatization, College of Teacher Education, Wenzhou University, Wenzhou 325035, China
* Correspondence: apzhao@wzu.edu.cn

Received: 12 January 2019; Accepted: 11 February 2019; Published: 21 February 2019

Abstract: To facilitate product developers capturing the varying requirements from users to support their feature evolution process, requirements evolution prediction from massive review texts is in fact of great importance. The proposed framework combines a supervised deep learning neural network with an unsupervised hierarchical topic model to analyze user reviews automatically for product feature requirements evolution prediction. The approach is to discover hierarchical product feature requirements from the hierarchical topic model and to identify their sentiment by the Long Short-term Memory (LSTM) with word embedding, which not only models hierarchical product requirement features from general to specific, but also identifies sentiment orientation to better correspond to the different hierarchies of product features. The evaluation and experimental results show that the proposed approach is effective and feasible.

Keywords: features prediction; sentiment analysis; LSTM

1. Introduction

The product feature requirements are the descriptions of the features and functionalities of the target product. This provides a way for a product developer to get and understand the expectations of users of the product. Product feature systems are undergoing continuing changes and a continuous updating process to satisfy users' requirements. Such requirements' evolution plays an important role in the lifetime of a product system in that they define possible changes to product feature requirements, which are one of the main issues that affect development activities, as well as product features.

However, the requirements of real product are often hidden and implicit. Product developers need to gather varying requirements from the users to know their ideas on what the product should provide and which features they want the product to include. Without the ability to discover these changing feature requirements from the users, they fail to make decisions about what features the product should provide to meet users' requirements. More importantly, in a real application, we need to identify the feature requirements of given products and the opinions being expressed towards each feature at different granularities from users. Taking smartphones as an example, some users care about general features such as appearance and screen size, while others may pay more attention to more specific features such as the user interface and running speed of the processor. On the one hand, different requirement information is interested in different granularities of features of a product. Analysis of specific product features and sentiments at a single granularity cannot capture the requirements of all users. We need to capture such different granularity requirements from user reviews. Therefore, it is necessary to convey the specific features and sentiments at different granularities of products to product developers so that they can easily understand the different levels of user concerns. On the other hand, the sentiment polarities of many words depend on the specific feature. For example, the sentiment “fast” is positive for the feature “processor”, but it would be negative for describing the feature “battery” of a smartphone. Therefore, it is more challenging to model feature granularity and to incorporate sentiment.

Sentiment analysis is the computational study of people’s opinions and attitudes towards entities such as products and their attributes. Understanding and explicitly modeling hierarchical product features and their sentiment are an effective way to capture different users’ requirements. In this paper, we demonstrate the hypothesis for the task of hierarchical product features requirement evolution prediction by sentiment analysis with a recurrent neural network and the Hierarchical Latent Dirichlet Allocation (HLDA). It presents a hierarchical structure of features and sentiments to product developers or requirement engineers so that they can easily navigate the different requirement granularities. From the viewpoint of product developers, it is easy to capture the requirement information of different users on different granularities of product features. They can understand what users think of each feature of the product application by taking both the features and sentiment into consideration.

The approach is explored to discover and predict the evolution of different granularity feature requirements of the product from the users’ review text. The approach combines deep Long Short-term Memory (LSTM) with the HLDA to predict product feature requirements of different hierarchies. The different hierarchical product features with their sentiment could be presented to a product developer as a detailed overview of the users’ primary concerns with the information about which product features are positive and which features are negative, which is used to suggest changes in product requirements from users. It can be used to decide whether or not the feature is a candidate for a product requirement change. A product developer could easily obtain an overview of the product features from the whole set of review text by navigating between the different hierarchies. This could help them decide on the necessary changes to the product feature requirements.

The main contributions of this paper are as follows:

- We propose considering the hierarchical dependencies of features and their sentiments for product feature requirement analysis by combining the power of supervised deep RNN with unsupervised HLDA. This shows the potential advantage of a hybrid approach of a joint deep neural network and topic model;
- We use a hierarchical structure of product feature representations with sentiment identification to capture product feature requirements of different granularities;
- We investigate the effectiveness of hierarchical product feature requirements prediction over different pre-trained word embedding representations and conduct extensive experiments to demonstrate their effectiveness.

The rest of this paper is organized as follows: In Section 2, we review related work. The hybrid text analysis approach for product feature requirements evolution prediction is described in Section 3. We indicate the effectiveness of the presented approach by evaluation and experimental results in Section 4. The discussion is provided in Section 5. Finally, we summarize our conclusions and future work in Section 6.

2. Related Work

This work focuses on methods for requirements evolution prediction through sentiment analysis of users’ online reviews. Several approaches have been studied to perform product feature requirements evolution detection effectively. We group related work into two categories: requirements evolution prediction through sentiment analysis and deep learning approaches for sentiment analysis.

2.1. Requirements Evolution Prediction through Sentiment Analysis

A number of studies have focused on requirement extraction for mobile applications [1]. Rizk et al. investigated users’ reviews’ Sentiment Analysis (SA) in order to extract user requirements to build new applications or enhance existing ones [2]. The authors analyzed users' feedback to extract the features and sentiment score of a mobile app in order to find useful information for app developers [3]. Guzman et al. presented an exploratory study of Twitter messages to help requirements engineers better capture user needs information for software evolution [4]. Jiang et al. used text mining technology to evaluate quantitatively the economic impact of user opinions that might be a potential requirement of software [5]. Guzman and Maalej proposed an automated approach to extract the user sentiments about the identified features from user reviews. The results demonstrated that machine learning techniques have the capacity to identify relevant information from online reviews automatically [6]. The analysis of user satisfaction is useful for improving products in general and is already an integral part of requirements evolution. An approach was presented for mining collective opinions for comparison of mobile apps [7]. To determine important changes in requirements based on user feedback, a systematic approach was proposed for transforming online reviews to evolutionary requirements [8]. A text mining method was proposed to extract features from precedents, and a text classifier was applied to classify judgments according to sentiment analysis automatically [9]. The authors proposed incorporating customer preference information into feature models based on sentiment analysis of user-generated online product reviews [10]. A semi-automated approach was proposed to extract phrases that can represent software features from software reviews as a way to initiate the requirement reuse process [11].

2.2. Sentiment Analysis with Deep Learning

Deep learning has been used for sentiment analysis because of its ability to learn high-level features and to find the polarized opinion of the general public regarding a specific subject automatically [12]. RNNs have been used for sentiment analysis by explicitly modeling dependency relations within certain syntactic structures in the sentence [13]. Besides these, Araque et al. attempted to improve the performance of sentiment analysis by integrating deep learning techniques with traditional surface approaches [14]. A knowledge-based recommendation system (KBRS) was presented to detect users with potential psychological disturbances based on sentiment analysis [15]. Irsoy and Cardie implemented a deep recurrent neural network for opinion expression extraction [16]. The work proposed a tree-structured LSTM and conducted a comprehensive study of using LSTM in predicting the sentiment classification [17]. Liu et al. proposed combining the standard recurrent neural network and word embeddings for aspect-based sentiment analysis [18]. Wang et al. introduced the Long Short-Term Memory (LSTM) recurrent neural network for twitter sentiment classification [19]. The latest research shows that attention and the dynamic memory network have become an effective mechanism to obtain superior results in sentiment classification task [20,21].

A number of research works have been proposed for supporting requirements evolution analysis. Previous research attempted to explore a machine learning model to discover software features and their sentiment simultaneously, which captures both semantic and sentiment information about software requirements encoded in users' reviews. However, our work differs from them mainly because we consider the dependencies of features and their sentiments. Most of the existing work so far has ignored the hierarchical structure of the features and sentiments, which assumes that all features are independent of each other. Although these existing methods are simple and commonly used, they suffer from the problem of feature-dependent sentiment identification from general to the specific. Knowledge about the relations between features and differentiation of the prior and contextual polarity of each feature are crucial. In order to better address this phenomenon, we convey it to a hierarchical structure in feature requirement analysis by using the combination of deep RNN and unsupervised hierarchical topic models. The joint model is able to capture inter-features and sentiments' relations without relying on any hand-engineered features or sentiment lexica.

3. Joint Text Analysis Model

3.1. Basic Framework

In order to model feature granularity and discover different requirements of users based on sentiment, we present a hierarchical feature requirement prediction method based on RNN and HLDA to solve the problem of hierarchical feature-dependent sentiment discovery. Figure 1 shows the joint model framework to present the basic working mechanism.

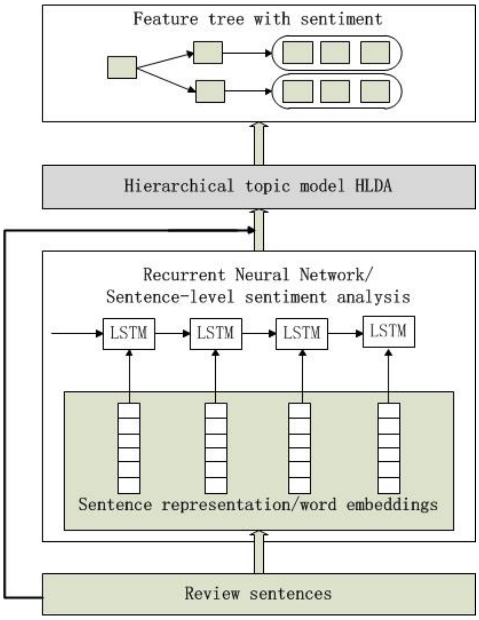
Feature tree with sentiment
Hierarchical topic model HLDA
Recurrent Neural Network/
Sentence-level sentiment analysis
LSTM
LSTM
LSTM
LSTM
Sentence representation/word embeddings
Review sentences

Figure 1. The joint model framework (LSTM+HLDA).

We introduce a recurrent neural network-based architecture that is able to learn both the feature-sentiment topics and the hierarchical structure from an unlabeled product review corpus. First, we use a component to identify sentence-level sentiment orientation by using an LSTM-based Recurrent Neural Network (RNN) with word embeddings. Intuitively, knowledge about the sentiment of surrounding words and sentential context should inform the sentiment classification of each review sentence. The sentiment polarity depends on the specific feature and context. This recurrent neural network-based component is able to consider intra-sentence relations to capture sentiment orientation.

Next, the second component is designed to identify hierarchical features, where these sentences with sentiment are fed to HLDA to discover hierarchical product features from general to specific to represent multi-granularity feature requirements. It is desirable to examine more fine-grained feature-based sentiment analysis of a product in more detail. This focuses on incorporating sentiment and modeling feature hierarchies to propagate the polarity of product features from general to specific. Thereafter, hierarchical requirement feature structure representation is generated with the corresponding sentiment, which contains more semantic information.

3.2. LSTM-Based Sentiment Analysis

In this subsection, we use LSTM-based sentiment analysis technology to identify the user's opinions being expressed towards each product feature. LSTM is a powerful type of recurrent neural network, which is to capture intra-sentence relations to provide additional valuable information for sentiment prediction. In our problem, we intend to use pre-trained word embedding in the LSTM-based RNN framework for the sentence-level sentiment classification task, which forms the basis for the next step to identify hierarchical features with sentiment orientation.

In the prevailing architecture of LSTM networks, there are an input layer, a recurrent LSTM layer, and an output layer. In the input layer, each word from sentences is mapped to a D-dimensional vector, which is initialized by pre-trained word embedding vectors. Given an input sequence of sentence $s = (features, sentiments)$, compositional context vector x_t is created by mapping each word token in s to a D-dimensional vector. This is designed to capture context dependencies between the features and sentiment of the sentence. The input layer can be represented as:

$$X = [h_{t-1}, x_t] \tag{1}$$

where the h_t is the vector of the hidden layer.

The LSTM layer connects to the input layer. The high-level compositional representations are learned by passing through the concatenated word embedding vector of the input layer to LSTM layers. LSTM incorporates "gates", which are neurons that use an activation function to remember certain input adaptively and are multiplied with the output of other neurons. It is capable of capturing relations among words and remembering important information for sentiment prediction. The LSTM-based recurrent architecture we used in this work is described as follows:

$$f_t = \sigma(W_f \cdot X + b_f) \tag{2}$$

$$i_t = \sigma(W_i \cdot X + b_i) \tag{3}$$

$$o_t = \sigma(W_o \cdot X + b_o) \tag{4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_c \cdot X + b_c) \tag{5}$$

where f_t, i_t, o_t, c_t are the forget gate, input gate, output gate, and memory cell, respectively. W is the weighted matrices. b is the bias of LSTM to be learned during training. σ is the sigmoid function. $\odot$ is element-wise multiplication.

The cell output units are connected to the output layer of the LSTM network for sentiment classification using softmax. We put the last hidden vector h_t of the representation of a sentence in a softmax layer.

$$h_t = o_t \odot tanh(c_t) \tag{6}$$

The probability of softmax sentiment classification in the output:

$$p(y_t = k|s, \theta) = \frac{exp(W_k^T h_t)}{\sum_{k=1}^{K} exp(W_k^T h_t)} \tag{7}$$

where w_k are the weights between the last hidden layer and the output layer. For sentence-level sentiment analysis, we use the Negative Log Likelihood (NLL) to fit the models from the training data by minimizing its value for the sentence sequence.

$$L(\theta) = \sum_{t=1}^{T} \sum_{k=1}^{K} y_{tk} \log p(y_t = k|s, \theta) \tag{8}$$

where $y_{tk} = I(y_t = k)$ is an indicator variable. The loss function minimizes the cross-entropy between the predicted distribution and the target distribution.

Based on such an LSTM recurrent architecture, we can capture sentiment information in the review sentence and identify polarities of sentiments.

3.3. Hierarchical Features Discovery

Hierarchical features discovery is to show a detailed overview of users' primary concerns with the information about which specific product features are positive or negative. For our hierarchical features and sentiments discovery of a product, we use the tree-structure model of HLDA [22].

Intuitively, a review often contains multiple product features and sentiments. Therefore, HLDA is reasonable to model a review text from a multiple product feature path. We feed review sentences with their sentiment from RNN into the HLDA to capture different granularity requirement features with sentiment orientation. The main motivation is to get the hierarchical product features and the feature-dependent sentiment, which is based on the assumption that a corpus of product reviews contains a latent structure of features and sentiments that can naturally be organized into a hierarchy. Formally, consider the infinite hierarchical tree defined by the HLDA to extract both the structure and parameters of the hierarchical tree automatically. The output of the model is a distribution over pairs of sentiment-feature for each sentence in the data. We employ the generative process as follows:

1 Draw the global infinite topic tree $T \sim rCRP(\gamma)$.
2 For each node Φ_k of feature k and sentiment terms in the tree T, draw a word distribution $\phi_k \sim Dirichlet(\beta)$.
3 For each sentence i in review document d:

    (1) Draw a feature-sentiment node $c \sim T$.
    (2) Draw a subjectivity distribution $\theta \sim Beta(\alpha)$.
    (3) For each word j

        i. Draw a word subjectivity $p \sim Binomial(1, \theta)$.
        ii. Draw the word $w \sim Multinomial(\phi_{k,p})$.

where $rCRP$ is the recursive Chinese restaurant process. This generative process defines a probability distribution across possible review corpora. Each feature and sentiment topic is a multinomial distribution over the whole vocabulary. All the feature and sentiment topics within a node Φ_k are independently generated from the same Dirichlet (β) distribution, which means they share the same general semantic theme. Subjectivity and non-subjective for each word indicate whether the word has sentiment orientation or not. With the extension of the HLDA model, the task of inference is to perform posterior learning to invert the generative process of review documents described above to estimate the hidden requirement features of a review collection. We use the collapsed Gibbs sampling method to approximate the posterior for the hierarchical feature sentiment identification model. The routine procedure of collapsed Gibbs sampling has been used in the Hierarchical Aspect Sentiment Model (HASM) [23], which solves a similar problem to our work in nature. Different from HASM, in our problem, we only need sample $p(c, p|w, \alpha, \beta, \gamma)$ for the feature and sentiment node c of each sentence and the subjectivity p of each word in a sentence.

The probability of generating sentence i in review document d from subjectivity p and node Φ_k is:

$$p(w_{di}|p, \Phi_k, \beta) \propto \prod_{l=0}^{1} \left(\frac{\Gamma(n_{k,l,-i}^{w,(.)} + \hat{\beta}_{l_i})}{\prod_w \Gamma(n_{k,l,-i}^{w,(w)} + \beta_{l_i,w})} \times \frac{\prod_w \Gamma(n_{k,l}^{w,(w)} + \beta_{l_i,w})}{\Gamma(n_{k,l}^{w,(.)} + \hat{\beta}_{l_i})} \right) \tag{9}$$

where $n_{k,l,-i}^{w,(v)}$ is the number of words having v_{th} vocabulary assigned to feature k and subjectivity l. n_{-i} represents the counter variable excluding index i.

Subjectivity sampling for each word is similar to the Gibbs sampling process in a basic LDA model with two topics: {0, non-subjective; 1, subjective}.

$$p(p_{dij} = k|w, p, c, \beta) \propto (n_{d,i,-j}^{p,(k)} + \alpha) \times \frac{n_{c_{d,i},k,-j}^{w,(v)} + \beta_{k,v}}{\sum_{r=1}^{V} n_{c_{d,i},k,-j}^{w,(r)} + \hat{\beta}_k} \tag{10}$$

where $n_{d,i,-j}^{p,(k)}$ represents the number of k-subjective words in sentence i of review document d. n_{-j} represents the counter variable excluding index j.

In this basic extension of the HLDA framework, it learns the hierarchical structure of product features from review text. Not only topics that correspond to requirement features are modeled, but also the features are retrieved to better correspond to the sentiments. In this way, the resulting hierarchical features and sentiment provide a well-organized hierarchy of requirement features and sentiments from general to specific. This provides easy navigation to the desired requirement feature granularity for product developers or requirement analysts. As a result, the joint hierarchical model is an effective way to provide meaningful feedback to the product developer and the requirements engineer. It provides the descriptive requirement information to product developers by the navigation of product features and the associated sentiments at different granularities, which helps developers decide whether or not the product feature is a candidate for a requirements change.

4. Evaluation and Experimental Results

4.1. Datasets

The dataset is from an existing publicly-available Amazon review datasets (http://jmcauley.ucsd.edu/data/amazon/). This dataset comprises product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996–July 2014. Reviews include product and user information, ratings, and a plain text review, which has already had duplicate item reviews removed and also contains more complete data/metadata. We selected the app for Android review dataset among these to use in our experiment. The final app review dataset comprises 752,937 total reviews. The datasets were pre-processed by separating sentences and removing non-alphabetic characters and single-character words.

4.2. Experimental Settings

LSTM-Based RNN:

We used a fixed learning rate of 0.01 and changed the batch size depending on the sentence length. We ran Stochastic Gradient Descent (SGD) for 30 epochs. We set the hidden layer sizes (h) as 50, 100, 150.

Pre-Trained Word Vectors:

We used the publicly-available word embedding Google News word2vec (http://word2vec.googlecode.com/svn/trunk/). Mikolov et al. proposed two log linear models for computing word embedding from large corpora efficiently: (i) a bag-of-words model, CBOW, which predicts the current word based on the context words, and (ii) a skip-gram model that predicts surrounding words given the current word. They released their pre-trained 300-dimensional word embedding (vocabulary size of 3 M) trained by the skip-gram model on the part of the Google News dataset containing about 100 billion words [24].

Hierarchical Topic Model:

The HLDA model was set up to discover hierarchical feature requirements with positive and negative sentiment from users' review sentences. The Dirichlet hyperparameter of the HLDA was set as α of 25.0, and the prior was set as γ of 0.01.

4.3. Evaluation Metrics

We seek to assess whether sentence-level sentiment analysis based on the joint deep neural network and hierarchical topic model could provide support for hierarchical product feature requirement evolutionary prediction. To evaluate the presented approach, we applied it on the review dataset and chose to compare different review text content analysis models for requirements evolution prediction to analyze of our approach experimentally. The evaluation comprises two parts. First, we evaluated the sentiment analysis technique using standard measurements, and second, we evaluated the relevance and the effect of feature-dependent sentiment of the automatically identified hierarchical product feature requirements from the reviews.

The accuracy of the sentiment classification results from our model framework was used to evaluate the quality of the results. The Area Under the Curve (AUC) is a common evaluation metric for binary classification problems. The AUC value is equal to the probability that a randomly-chosen positive example is ranked higher than a randomly-chosen negative example. We used AUC to show the change in accuracy of sentence-level sentiment analysis under our joint model framework.

We utilized a further metric to determine the quality of the multi-granularity product feature generated by the HLDA. Product feature requirements relevance assesses how well the extracted text topics represent real product features, which show that the topics contain information that helps understand, define, and evolve the feature requirements of the product. Feature-dependent sentiment classification accuracy evaluates how well the extracted hierarchical structure represents a product feature, and the children nodes represent the sentiment polarities associated with it.

We evaluated both the requirements relevance and feature-dependent sentiment classification using precision, recall, and the F-measure. A feature was defined as a true positive if it was extracted from an online user's review by HLDA and was also manually identified in that review. A feature was defined as a false positive if it was automatically associated with a review in a topic, but was not identified manually in that review. False negatives were features that were manually identified in a review, but were not present in any of the extracted topics associated with the review.

4.4. Experimental Results

The different sentiment analysis techniques on product review enabled us to compare the results of similar approaches to analyze the effectiveness of each. Recent extensive studies have shown that the RNN-based sentiment classification method outperforms the previously-proposed sentiment analysis model. However, we mainly focused on comparison between the text analysis approaches, which are applicable to the requirements evolution detection from the online reviews. The Aspect and Sentiment Unification Model (ASUM) was used on user comments of software applications for software requirements changing analysis [25], which incorporates both topic modeling and sentiment analysis of the same model. Our evaluation compared the results with the ASUM text analysis approach to requirement evolution on the datasets. For each experiment, we evaluated the sentiment classification of requirement features from review text. We intended to evaluate whether the resulting hierarchical product features sentiment polarities classification could accurately represent the true sentiment orientation. The approach proposed in this work classified the multi-granularity product features extracted from the review text into positive and negative. This sentiment classification information can be used to detect the unique product features with the positive and negative user approval. In our framework, hierarchical product features sentiment classification was done by LSTM-based RNN. By comparing the product features sentiment classification extracted by our approach with the ASUM, we aimed to justify the effectiveness of integrating the deep neural network with pre-trained word embedding into the product features sentiment identification process. Figure 2 presents the results of sentiment classification on the dataset.

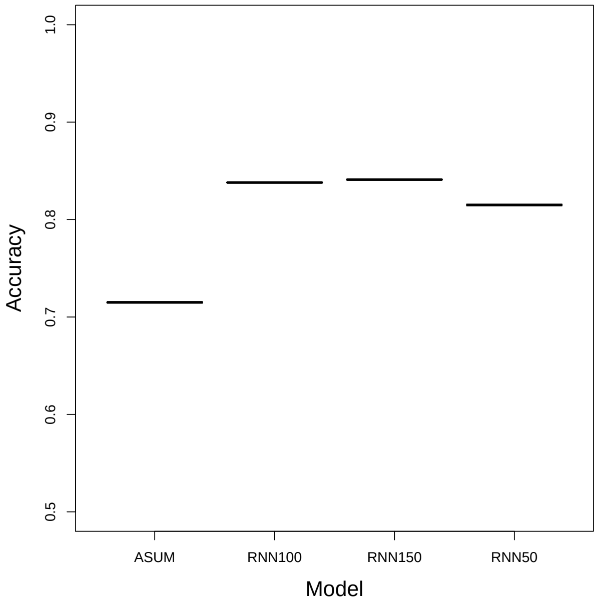

Figure 2. Accuracy comparison under the RNN with different numbers of hidden layers and the Aspect and Sentiment Unification Model (ASUM). RNN50, RNN with 50 hidden layers. RNN100, RNN with 100 hidden layers. RNN150, RNN with 150 hidden layers.

We can see that the relative accuracy of product features sentiment classification achieved using RNN with different numbers of hidden layers was better than ASUM. This may be mainly because of the feature sentiment classification methods having distinct characteristics. ASUM neither built any internal representations in a sentence nor took into account the structure of the sentence, while the RNN operated on sequences of words and built internal representations by word embedding before detecting the sentiment orientation.

We manually examined the 10 most popular extracted features for each of the 20 topics generated by HLDA to measure the feature requirements' relevance and feature-based sentiment classification. To compare the results of our approach with the manual analysis, we created a groundtruth set of product features referred to in the reviews and their associated sentiments. Following the previous studies, the content analysis techniques were used to create the groundtruth set [26]. Figure 3 shows the results of the relevance comparison under the joint model and ASUM.

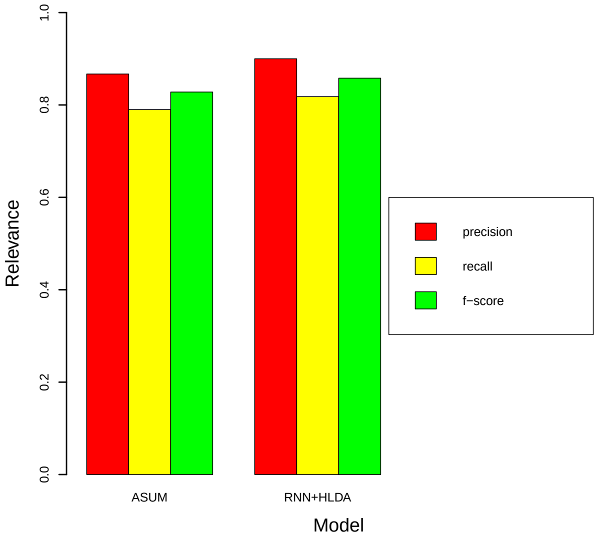

Figure 3. Relevance comparison under the joint model and ASUM.

We can note that the results were comparable for the two models. It can be seen that both of the models had good performance of relevance to requirements engineering for products. The extracted product features were usually words of topics describing actual product features. This is because both HLDA and ASUM identify the topics using models that are similar in nature, while the topics from the HLDA had different granularities.

The feature-dependent sentiment classification accuracy is shown in Figure 4.

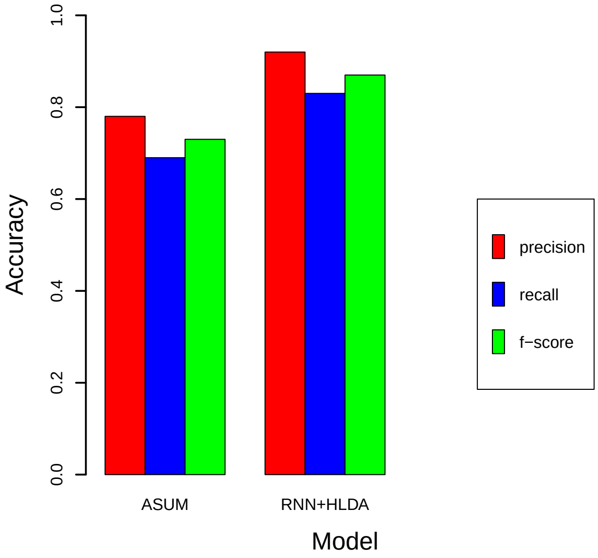

Figure 4. The feature-dependent sentiment analysis comparison under the joint model and ASUM.

We intended to evaluate whether the identified feature-sentiment hierarchy could show different sentiment orientations of some words depending on the specific feature at different granularities. The results of both the joint model and HASM show that the discovered feature-sentiment hierarchy can identify the sentiments for specific features and the associated opinions. We can see that the joint hierarchical model outperformed unsupervised HASM without any hand-engineered features. This may be mainly because of the distinct characteristics of sentiment classification. The LSTM-based recurrent neural network architecture was able to consider intra-sentence relations and provided valuable clues for the sentiment prediction task, while the HASM model, each feature or sentiment polarity as a distribution of words. This result confirms that the joint model was capable of analyzing sentiment for certain features at different granularities.

These automatically-identified features with their sentiment orientation are important for understanding how users evaluate the product, which might contain valuable information for the feature requirement evolution and maintenance of the product. To provide the experimental results as an example of the hierarchical product feature requirements presented in the user review, Table 1 is an example of the three-level hierarchical requirement features with sentiment orientation by combining RNN with HLDA.

Table 1. Hierarchical requirement features with sentiment orientation.

Hierarchy		Features and Sentiments
Level-0	features:	interface, update, power, usability, performance
	pos:	handy, could, pretty, friendly, usable
	neg:	frustrating, confusing, crash, fault, inconvenient
Level-1	features:	permission, log, option, connection, availability
	pos:	smart, install, graphical, adaptive, instant
	neg:	error, slow, boring, stupid, block
Level-2	features:	popup, buttons, text, account, login
	pos:	attractive, compatible, support, control, search
	neg:	small, missing, confusion, poor, useless

The result presented different granularity information for the hierarchical organization of features and sentiments. We can see that the produced hierarchical structure showed certain features and corresponding sentiments at various granularities. The hierarchical nodes showed the requirement features at various granularities with different feature-dependent sentiments. The sentiment word changed its polarity according to the specific feature. This hierarchical structure of feature-sentiment shows that general features became specific features of the product as the depth increased. The hierarchical features were relevant in the sense that they covered important features at different granularities of the product applications. The result confirms that the identified product requirement features were organized into a hierarchical structure from general to specific. Requirement feature-dependent sentiment polarities were obtained mainly because of RNN-based sentence-level sentiment classification being fed into the hierarchical topic model.

These evaluations and experimental results verify the effectiveness of the hierarchical feature requirements evolution prediction. The qualitative results show that the features automatically identified from review text described the overall requirement change of the products. As a result, we conclude that the combination of the supervised deep recurrent neural network and unsupervised hierarchical topic model is an effective approach to requirement evolution identification from users' review text. The above experimental results and analysis show that our proposed method is efficient, feasible, and practical.

5. Discussion

Our hierarchical joint model achieved results competitive with the software requirements changing analytical model of ASUM and the hierarchical aspect sentiment model of HASM. We highlighted the identification of hierarchical features and sentiments where this facilitates product feature requirements evolution predictions from users' reviews. In our work, hierarchical features requirement evolution prediction can be divided into two core components: supervised LSTM deep learning for sentence-level sentiment analysis and unsupervised topic modeling for hierarchical features identification. Thus, we analyzed the impact of different word embeddings, activation functions, and sizes of hidden layers for LSTM sentiment analysis and discussed a prior of the Dirichlet distribution for HLDA topic modeling.

Word embedding is a distributed representation of words, which is combined with a word and its context to represent a word as real valued, dense, and having low-dimensional vectors and greatly alleviated the data sparsity problem [24]. Syntactic or semantic properties of the word are potentially described by each dimension of word embedding. The semantic representation of sentences and texts was captured by the use of word embedding. Pre-trained embeddings improved the model's performance. Further, combining different pre-trained word embedding with deep neural networks brought new inspiration to various sentiment analysis tasks. We plugged the readily available embedding of sentences into the LSTM network framework and used it as the only feature to avoid manual feature engineering efforts for sentimental orientation classification. Polarity was predicted on either a binary (positive or negative) or multivariate scale using sentiment polarity classification techniques. The compositional distributed vector representations for review sentences were obtained by concatenating their word embedding representations. The resulting high-level distributed representations were used as the individual features to classify each sentence-sentiment pair. As a result, the LSTM network's performance can be improved by choosing the optimal hyperparameters of the dimension of pre-trained embeddings for the word and paragraph vector training.

The proper selection of the activation function is an effective way to initialize the weight matrix W to reduce the gradient vanishing and gradient explosion effect. The activation functions used in LSTM for sentiment analysis are sigmoid and tanh. The update of the state value is described as Equation (5) in the form of addition with the sigmoid and tanh function to solve the problem of long-term dependencies in LSTM better. The size of hidden layers in LSTM depends on the application domain and context. A model with many hidden layers will resolve non-linearly separable data. For the sentiment classification task with the deep neural network model, we assumed that the data required a non-linear technique. The introduction of the hidden layer(s) of deep LSTM network made it possible for the deep network to exhibit non-linear behavior. The different size of hidden layers in LSTM affected the results of the sentiment classification task. For hierarchical topic modeling, we made the observation that features descending from parent feature k must be more similar to parent feature k than features descending from other features, and there are more sparse features with increasing depth of feature k in a hierarchical structure. We used a prior of Dirichlet distribution β as a parameter to generate more sparse features with increasing depth of feature k of a hierarchical structure. The smaller the parameter value is, the more sparse the distribution is when the values of a parameter are less than one.

In light of the implicit expressions for product features and sentiment in the users' reviews, implicit semantic discovery is often preferred. Exploring better ways to incorporate such implicit information into word embedding representation, as well as methods to inject other forms of context information is thus an important research avenue, which we propose as future work.

6. Conclusions and Future Works

In this paper, we explored the use of the joint unsupervised topic model and deep LSTM-based sentiment analysis to identify hierarchical feature sentiment for requirement evolution prediction. The main motivation is to facilitate the product developers to capture and understand the requirements change in a navigable way between the different product feature requirement granularities. The results of this work significantly contribute to efforts toward automatic text mining analysis for product requirements engineering.

The methodology revolves around the sentiment analysis and hierarchical topic modeling technology to capture an overview of users' primary concerns. The approach was able to detect product features mentioned in the user review text for different granularities with sentiment orientation. The work presented here demonstrates the potential of combining deep neural network-based sentiment analysis with topic modeling for requirement evolution prediction from users' reviews. Our joint framework makes assumptions that product feature context dependent sentiment can be captured by learning compositional word embedding representations of corresponding features. Distributed word embedding can be different from the different training objectives and language models. Therefore, the quality of the word embedding could have an impact on the efficacy of the sentiment classification results.

Because of the many implicit expressions for product features and sentiment in the user's review, the text analysis-based approach to product requirements evolution detection should be adapted to the implicit context to identify implicit product features and sentiment. This is an issue worth investigating for our future work.

Author Contributions: Conceptualization, A.Z. and L.Z.; Methodology, A.Z. and L.Z.; Validation, L.Z.; Formal Analysis, A.Z. and L.Z.; Writing—Original Draft Preparation, L.Z.; Writing—Review & Editing, A.Z.

Funding: This research was funded by the Chongqing Research Program of Basic Research and Frontier Technology Grant Number cstc2018jcyjAX0708.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nagappan, M.; Shihab, E. Future trends in software engineering research for mobile apps. In Proceedings of the 23rd IEEE International Conference on Software Analysis, Evolution, and Reengineering, Osaka, Japan, 14–18 March 2016.
2. Rizk, N.M.; Ebada, A.; Nasr, E.S. Investigating mobile applications' re-quirements evolution through sentiment analysis of users' reviews. In Proceedings of the 11th International Computer Engineering Conference (ICENCO 2015), Cairo, Egypt, 29–30 December 2015; pp. 123–130.
3. Guzman, E.; Maalej, W. How do users like this feature? a fine grained sentiment analysis of app reviews. In Proceedings of the 2014 IEEE 22nd International Requirements Engineering Conference (RE), Karlskrona, Sweden, 25–29 August 2014; pp. 153–162.
4. Guzman, E.; Alkadhi, R.; Seyff, N. An exploratory study of Twitter messages about software applications. *Requir. Eng.* **2017**, *22*, 387–412. [CrossRef]
5. Jiang, W.; Ruan, H.; Zhang, L. *Analysis of Economic Impact of Online Reviews: An Approach for Market-Driven Requirements Evolution*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 45–59.
6. Guzman, E.; Alkadhi, R.; Seyff, N. A needle in a haystack: What do twitter users say about software? In Proceedings of the 2016 IEEE 24th International Requirements Engineering Conference (RE), Beijing, China, 12–16 September 2016.
7. Malik, H.; Shakshuki, E.M. Mining collective opinions for comparison of mobile apps. *Procedia Comput. Sci.* **2016**, *94*, 168–175. [CrossRef]
8. Jiang, W.; Ruan, H.; Zhang, L.; Lew, P.; Jiang, J. For user-driven software evolution: requirements elicitation derived from mining online reviews. In Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Tainan, Taiwan, 13–16 May 2014; pp. 584–595.
9. Liu, Y.H.; Chen, Y.L. A two-phase sentiment analysis approach for judgement prediction. *J. Inform. Sci.* **2018**, *44*, 594–607. [CrossRef]
10. Zhou, F.; Jiao, J.R.; Yang, X.J.; Lei, B. Augmenting feature model through customer preference mining by hybrid sentiment analysis. *Expert Syst. Appl.* **2017**, *89*, 306–317. [CrossRef]
11. Bakar, N.H.; Kasirun, Z.M.; Salleh, N.; Jalab, H.A. Extracting features from online software reviews to aid requirements reuse. *Appl. Soft Comput.* **2016**, *49*, 1297–1315. [CrossRef]
12. Rojas-Barahona, L.M. Deep learning for sentiment analysis. *Lang. Linguist. Compass* **2016**, *10*, 701–719. [CrossRef]
13. Zhang, L.; Wang, S.; Liu, B. Deep learning for sentiment analysis: A survey. *WIREs Data Min. Knowl. Discov.* **2018**, *8*, e1253. [CrossRef]
14. Araque, O.; Corcuera-Platas, I.; Sanchez-Rada, J.F.; Iglesias, C.A. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Syst. Appl.* **2017**, *77*, 236–246. [CrossRef]
15. Rosa, R.L.; Schwartz, G.M.; Ruggiero, W.V.; Rodriguez, D.Z. A Knowledge-Based Recommendation System that includes Sentiment Analysis and Deep Learning. *IEEE Trans. Ind. Inform.* **2018**. [CrossRef]
16. Irsoy, O.; Cardie, C. Opinion mining with deep recurrent neural networks. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 720–728.
17. Tai, K.S.; Socher, R.; Manning, C.D. Improved semantic representations from tree-structured long short-term memory networks. *arXiv* **2015**, arXiv:1503.00075. Available online: https://arxiv.org/abs/1503.00075 (accessed on 13 February 2019).
18. Liu, P.; Joty, S.; Meng, H. Fine-grained opinion mining with recurrent neural networks and word embeddings. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1433–1443.

19. Wang, X.; Liu, Y.; Chengjie, S.U.N.; Wang, B.; Wang, X. Predicting polarities of tweets by composing word embeddings with long short-term memory. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Beijing, China, 27–31 July 2015; pp. 1343–1353.
20. Zhang, Y.; Er, M.J.; Venkatesan, R.; Wang, N.; Pratama, M. Sentiment classification using Comprehensive Attention Recurrent models. In Proceedings of the International Joint Conference on Neural Networks (IJCNN 2016), Vancouver, BC, Canada, 24–29 July 2016; pp. 1562–1569.
21. Kumar, A.; Irsoy, O.; Ondruska, P.; Iyyer, M.; Bradbury, J.; Gulrajani, I.; Zhong V.; Paulus, R.; Socher, R. Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. In Proceedings of the International Conference on Machine Learning (ICML2016), New York, NY, USA, 19–24 June 2016; pp. 1378–1387.
22. Kim, J.H.; Kim, D.; Kim, S.; Oh, A. Modeling topic hierarchies with the recursive Chinese restaurant process. In Proceedings of the 21st ACM International Conference on Information and Knowledge Management, Maui, HI, USA, 29 October–2 November 2012; pp. 783–792.
23. Kim, S.; Zhang, J.; Chen, Z.; Oh, A.H.; Liu, S. A hierarchical aspect-sentiment model for online reviews. In Proceedings of the 27th AAAI Conference on Artificial Intelligence, Bellevue, WA, USA, 14–18 July 2013; pp. 526–533.
24. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 3111–3119.
25. Galvis Carreño, L.V.; Winbladh, K. Analysis of user comments: An approach for software requirements evolution. In Proceedings of the 2013 International Conference on Software Engineering, San Francisco, CA, USA, 18–26 May 2013; pp. 582–591.
26. Maalej, W.; Robillard, M.P. Patterns of Knowledge in API Reference Documentation. *IEEE Trans. Softw. Eng.* **2013**, *39*, 1264–1282. [CrossRef]