Research on Topic Knowledge Element Extraction Methods in Professional Social Media: Postprint

Authors: Lin Jie, Miao Runsheng, Zhang Zhenyu

Date: 2023-07-26T00:00:00+00:00

Abstract

[Purpose/Significance] Using automotive forums as a case study, this paper proposes a method for extracting thematic knowledge elements from professional social media text. [Methods/Procedures] Firstly, the LDA model is used to extract topics from texts in automotive forums, and deduplication is performed to form a topic list. Secondly, based on the T-LSTM model, a deep learning model that integrates topic features, a sentiment analysis model suitable for automotive forum texts is constructed. Then, by calculating the importance of each word in the TextRank graph model and the Word2Vec topic similarity of each word, sentiment keywords and key sentences are extracted to interpret and supplement text themes and sentiment tendencies. Finally, the above methods are integrated to output structured thematic knowledge elements. [Results/Conclusions] Experimental results show that the qualification rate of extracted thematic knowledge elements reached 69.1%, indicating that the proposed thematic knowledge element extraction method can relatively accurately extract knowledge elements centered on knowledge themes, achieving structured transformation of knowledge.

Full Text

Preamble

Research on Extraction Methods of Topic Knowledge Tuples in Professional Social Media

Lin Jie, Miao Runsheng, Zhang Zhenyu
School of Economics and Management, Tongji University, Shanghai 200092

Abstract

[**Purpose/Significance**] Taking automotive forums as an example, this paper proposes a method for extracting topic knowledge tuples from professional social media text. [**Method/Process**] First, the LDA model is used to extract topics from automotive forum texts, and duplicate topics are removed to form a topic list. Second, a deep learning model T-LSTM that integrates topic features is constructed to build a sentiment analysis model suitable for automotive forum texts. Then, by calculating the importance of each word in the TextRank graph model and the Word2Vec topic similarity of each word, sentiment keywords and key sentences are extracted to explain and supplement the text topics and sentiment orientation. Finally, the above methods are integrated to output structured topic knowledge tuples. [**Result/Conclusion**] Experimental results show that the qualification rate of extracted topic knowledge tuples reaches 69.1%, indicating that the proposed method can accurately extract knowledge elements around knowledge themes and achieve structured transformation of knowledge.

Keywords: topic knowledge tuple; topic extraction; long short-term memory neural network; sentiment analysis
Classification Number: G202
DOI: 10.13266/j.issn.0252-3116.2019.14.012

Knowledge tuples, also known as knowledge units or knowledge elements, are fundamental knowledge primitives used to operate and manage knowledge— independent knowledge units that can be freely segmented, expressed, accessed, organized, retrieved, and utilized. Topic knowledge tuples represent one form of knowledge tuple expression, where elements include knowledge topic words and key information related to the topic. Since topic words can accurately reflect various implicit effective associations between knowledge tuples, such as hierarchical relationships, parallel relationships, and cluster relationships, topic knowledge tuples are considered an appropriate form of knowledge expression. This paper defines professional social media as a content production and exchange platform where internet users share and exchange opinions, insights, experiences, and perspectives on specific professional matters. Professional social media is a special type of social media, typically in the form of professional forums or communities, such as "Autohome," "Xiaomi Community Official Forum," and "Hupu NBA Forum." Compared with other social media like Weibo and Facebook, professional social media contains more professional content and longer texts. Addressing the characteristics of professional social media corpora—massive volume, variable length, casual creation, and colloquialism—and the needs for knowledge operation and management, this paper proposes a method that uses text titles and content as data sources to extract topic knowledge tuples with the structure of "text topic, topic sentiment orientation, topic keywords, topic key sentences."

Topic knowledge tuples organically combine knowledge management and mod-

ern information technology, drawing on concepts from knowledge management such as tacit knowledge classification, knowledge refinement, and knowledge application, while employing technologies like big data processing, text mining, and machine learning. They hold high application value in numerous fields. Extracting topic knowledge tuples from massive texts enables retrieval, free operation, and management of knowledge content itself, while transforming the unit of knowledge control from documents to topic knowledge tuples, thereby improving the efficiency and flexibility of knowledge retrieval and operation. Utilizing the correlation between topics in topic knowledge tuples enables knowledge reorganization and creation, as well as quantification and evaluation of knowledge. Furthermore, in professional social media, massive user comment data contains rich user innovation knowledge. Realizing the extraction of topic knowledge tuples from professional social media corpora can refine high-value information from massive comment data, reducing the difficulty and cost of knowledge acquisition. The sentiment orientation, keywords, and key sentences in professional social media topic knowledge tuples can provide a data foundation for measuring and monitoring current topic popularity. Topic knowledge tuple extraction from professional social media forms the basis for various knowledge management and innovation activities. Enterprises can use topic knowledge tuples for customer demand mining and collaborate with users on interactive innovation, while governments and academic institutions can use them for social public opinion simulation research, 梳理 social opinion themes, monitor sudden public opinion events, and provide a basis for formulating public opinion governance measures and regulation strategies.

2 Related Research

The innovations of this paper include: (1) Since topics are the ideological core of professional social media text content, this paper designs a topic-centered knowledge tuple structure for professional social media texts, including text topic, topic sentiment, and topic keywords/sentences; (2) Addressing the characteristics of professional social media user comment corpora—massive volume, uneven user knowledge levels, variable text length, mixed and low-quality content, professional yet colloquial vocabulary—this paper proposes a method and technical route for extracting topic knowledge tuples and conducts experimental verification; (3) Due to the massive volume of professional social media corpora, ensuring both quality and speed of topic knowledge tuple extraction is essential. This paper introduces deep learning technology into knowledge tuple extraction. The rapid development of deep learning applications in text mining in recent years, with significant progress in sentiment analysis, text similarity calculation, and other tasks, provides guarantees for the quality and speed of topic knowledge tuple extraction.

In the field of knowledge tuple extraction research, Wen Youkui et al. defined and classified the content of knowledge tuples and described and implemented extraction schemes for literature resources, with a knowledge tuple structure

including three elements: "type, name, and content." However, this knowledge tuple structure is simplistic and lacks refinement of valuable information. This paper designs a knowledge tuple structure encompassing topics, sentiment, and keywords/sentences, which not only extracts knowledge content but also refines implicit knowledge contained in texts, such as topics and sentiment orientations. Jiang Yongchang described a conversion framework from text entity layer to semantic layer to knowledge unit layer based on the knowledge grid architecture, constructing a knowledge evolution framework from theoretical and technical perspectives, but did not specifically implement it. This paper systematically describes the extraction method for topic knowledge tuples and verifies its effectiveness through experiments. Liu Miao et al. proposed a knowledge tuple extraction method for literature resources based on LSTM joint modeling, which improved sentiment analysis accuracy and achieved product attribute extraction from product review corpora, forming knowledge tuples with the structure of "product, product attribute, sentiment orientation." However, knowledge tuple extraction methods for short texts like Weibo cannot adapt to the long texts and professional vocabulary in professional social media. This paper adopts deep learning-based word embedding methods to support topic sentiment and keyword/sentence extraction, better adapting to the text characteristics of professional social media. Y. Yin et al. found that product reviews and other additional information (such as user and product information) are helpful for joint classification modeling using neural networks for sentiment analysis. Based on this, this paper uses text and text topics as features, employing neural network joint classification modeling to improve sentiment analysis accuracy.

In terms of text category extraction, the main extraction objects in the above research are academic literature resources, which generally have clear topic classifications and keywords, making duplicate extraction of topics and keywords unnecessary. This paper constructs methods for extracting topics and topic keywords/sentences for professional social media texts. Yang Liang proposed a method for Sina Weibo texts using sentence-internal and global information fusion, achieving sentence-level knowledge tuple extraction by calculating similarity between sentences. However, topic sentence extraction only considers topics within single documents. This paper considers global, document-level topic elements when extracting topics, reducing redundancy in extracted topics.

3 Research Methods

3.1 Research Framework

This paper defines topic knowledge tuples as follows: In a professional social media corpus D containing M articles, a topic knowledge tuple u is extracted from the title h and content c of article m, structured as <text topic t, topic sentiment orientation p, keywords kw, key sentences ks>, i.e., u: (t, p, kw, ks).

The research framework is shown in Figure 1 [Figure 1: see original paper].

Specifically: (1) Crawl user comment texts from professional social media to build a user corpus; (2) Use the LDA model for topic extraction and merge duplicate topics to obtain a topic model and global topic list T, which provides the thematic foundation for subsequent topic sentiment analysis and topic keyword/sentence extraction; (3) Use the LDA model to label post topic polarity and simultaneously perform sentiment labeling to construct a T-LSTM-based sentiment analysis model that outputs sentiment orientation p; (4) Based on the TextRank algorithm and Word2Vec topic word similarity algorithm, calculate the weighted importance of keywords and sentences to extract sentiment keywords and key sentences; (5) Integrate the above models, train and encapsulate the extraction method for topic knowledge tuple u, and conduct experimental analysis and verification.

3.2 Topic Extraction Model

This paper first trains an LDA model to mine an appropriate number of topics from the professional social media corpus D, obtaining a topic list T. The LDA topic model can extract a global topic list from the professional social media corpus (see Table 2). By inputting each document individually into the LDA model, we obtain the corresponding topic for each document in the corpus. This model provides the thematic foundation for topic-oriented sentiment analysis and keyword/sentence extraction.

3.2.1 Building the LDA Topic Extraction Model The main idea of the LDA model is to find the distribution of documents over topics and the distribution of topic words over topics, i.e., each document corresponds to one or more topics, and each topic has multiple topic words. The core steps are: count the topics of words in each document to obtain the document-topic distribution (see formula (1)), count the distribution of topic words in the corpus to obtain the topic word distribution ϕ in the LDA topic (see formula (2)).

Formula (1):

$$\theta_{mk} = \frac{n_{mk,-i} + \alpha_k}{\sum_{s=1}^{K}(n_{ms,-i} + \alpha_s)}$$

Formula (2):

$$\phi_{kt} = \frac{n_{kt,-i} + \eta_t}{\sum_{s=1}^{T}(n_{ks,-i} + \eta_s)}$$

In formulas (1) and (2), K is the number of topics; α is the hyperparameter of the _m distribution, representing the relative strength between topics, which is a K-dimensional vector, and αk *is the k-th element of* α*; is the hyperparameter of the* ϕk distribution, which is a T-dimensional vector, where T is the dictionary size. In formula (1), n*{mk,-i} is the number of words assigned to topic k in article m, excluding the current word i. In formula (2),* n{kt,-i} is the number of words assigned to word t in topic k, excluding the current word i.

From the topic distribution , the top n topics with the highest distribution are selected as the initial topic list T_0. From the topic word distribution ϕ, the top m topic words with the highest distribution for each topic are selected as the topic words t in the topic list.

3.2.2 Topic Deduplication The topics extracted by LDA may contain duplicate or redundant topics. After obtaining the initial topic list T_0, this paper performs topic deduplication by calculating topic similarity to obtain the topic list T for corpus D.

To calculate whether two topics are duplicate or redundant, we first need to calculate the similarity between two topics. Extract the top W words from two topics to form sets A and B, respectively, then calculate the Jaccard Similarity of the two sets using the following formula:

$$J(A,B) = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Traverse the topics in topic list T, calculate the similarity between two topics using formula (3); compare the similarity J(A,B) of each topic pair with a given threshold (JaccardThreshold), record all topic pairs with similarity greater than the threshold, and finally merge them using the Disjoint-Set method to obtain topic list T. After topic deduplication, the topic distribution becomes more reasonable, manual naming workload is reduced, and extraction speed and quality are improved. Ultimately, the topic list T and corresponding topic words t belonging to corpus D are extracted.

3.3 Topic Sentiment Analysis Model

Understanding user sentiment orientation toward topics in text is one of the key tasks in knowledge tuple extraction. For professional social media text, this task can help knowledge users understand user needs and quantify user evaluations. Based on topics extracted by the LDA model, this paper uses an LSTM model to calculate the sentiment orientation of user-posted text (posts). Due to the recursive nature of the model, this paper uses both topic labels and sentiment labels for joint supervision during the LSTM model training process. Joint supervision can, on the one hand, extract sentiment orientation around text topics; on the other hand, it can utilize the correlation between topics and sentiment orientation to improve sentiment analysis accuracy.

3.3.1 Correlation Between Topic and Sentiment In professional social media, user statements are expressed in the form of posts. Due to variable post length and mixed categories, their sentiment attributes are difficult to grasp. However, posts in professional social media generally have distinct topics, and these topics often correlate with sentiment orientation. Taking automotive forums as an example, posts include topics such as “car purchase showcase,”

"faults and repairs," and "configuration comparison." After analyzing the correlation between topics obtained from the LDA model and the sentiment orientation of each post, we conclude that post topics and the sentiment orientation contained in posts have a strong correlation. The statistical results of sentiment orientation under each topic are shown in Table 1 .

Table 1 shows that posts on the car purchase showcase topic are mostly positive; posts involving fault, anomaly, and repair descriptions are mostly negative; while activity and social posts, and other categories such as second-hand transaction posts, mostly have no clear sentiment orientation. The statistics show that post topic is a strong correlation variable. Inputting it into the LSTM model as a feature can, on the one hand, make the sentiment orientation results closer to the post topic, and on the other hand, improve the model classification effect.

3.3.2 Topic-Enhanced LSTM Sentiment Classification Model This paper introduces topic information obtained in advance through the LDA method into the LSTM model, naming it the Topic-enhanced LSTM sentiment classification model (T-LSTM). The main idea of T-LSTM is: using the recursiveness of hidden layers in LSTM, input the topic word information from the LDA model as subsequent time steps into the model, then train using the sentiment orientation labels of samples on that topic. By learning the correlation between sentiment orientation and topic information, the accuracy of outputting sentiment orientation p on that topic is improved.

The overall structure of the T-LSTM model is shown in Figure 2 [Figure 2: see original paper], containing three layers of networks, from bottom to top: Embedding layer, LSTM layer (see Figure 3 [Figure 3: see original paper]), and MLP layer.

First layer, Embedding layer. The embedding layer is located at the bottom of the entire model. Its function is to reduce the dimensionality of word vectors processed by One-hot Vector, thereby reducing model complexity. The output y_i of word embedding serves as input to the learning model g: y→z, where the corresponding z_i value in task g is known. Through sample data {(x_i, z_i)}_{i=1}^N, the learning model k: x→z is trained, i.e., z = g(f(x)). The model y = f(x) in this process is the word embedding model.

Second layer, T-LSTM core network layer. Here, T represents the topic vector and P represents the sentiment orientation vector. The input to the entire LSTM layer is word vectors at different time steps (x_1, x_2, ..., x_t) and subsequent topic word vectors (T_1, T_2, ...), with the output being the vector corresponding to sentiment orientation P. After introducing topic information as a feature, the structure of the LSTM core layer is shown in Figure 3 [Figure 3: see original paper]. In Figure 3, each node represents a hidden layer containing a memory block, with each memory block's input being the output of the previous layer and the modified input. That is, the output of the post text layer serves as input to the topic feature layer, and the final output layer

includes both text information and topic information. In Figure 3, W_1 and W_2 are weight matrices for input and output vectors, and W_P are weight matrices for sentiment orientation P between hidden layers.

Third layer, Multi-layer Perceptron (MLP) network. Input the topic vector and sentiment orientation vector obtained from the second layer into the MLP layer. The vector output by the MLP layer passes through a Softmax layer to obtain the probability of sentiment orientation labels (P_k), where k represents sentiment orientation. Under the condition of model parameters obtained from training, the target probability can be expressed as:

$$p(P_k|x,\mu) = \frac{e^{W_P^k x + b_P^k}}{\sum_{i=1}^{|P|} e^{W_P^i x + b_P^i}}$$

Assuming training samples M and model nodes S, the loss function during training is defined as:

$$L(\mu) = \sum_{k=1}^{M} l\{P_k = j\} \times \log p(P_k|x,\mu) + \alpha|(\mu)|^2$$

In formula (5), l{P_k=j} indicates that if P_k=j holds, the value of l is 1, otherwise 0; $\alpha|(\ )|\hat{}2$ is the penalty term in the loss function, represents parameters trained in the model, α is the penalty coefficient with a value between [0,1].

This model uses the Adam algorithm for training, utilizing first and second moment estimates of gradients to dynamically adjust the learning rate for each parameter. Dropout technology is used here to prevent model overfitting, and the mini-batch method is employed for training. After training, the model is serialized and saved, and finally used to output the sentiment orientation p in topic knowledge tuples.

3.4 Topic Keyword and Sentence Extraction Model

The topics extracted by LDA are global topics obtained at the document collection level. For individual documents, their own topics often cannot correspond one-to-one with LDA topic words, resulting in redundant topic words. Document-level keywords and sentences differ from LDA topics—they extract key information from the document itself, with finer granularity than LDA topic words, and keywords originate from the document itself. This paper combines document-level keywords with LDA topic word extraction algorithms to extract document keywords and key sentences without deviating from major topics, providing interpretation and supplementation for LDA topics and sentiment orientation. Therefore, this paper uses the TextRank algorithm to calculate word and sentence importance in single documents, considering document-level keyword/sentence extraction; on the other hand, it uses the Word2Vec algorithm to calculate similarity between document words/sentences and document

topic words, considering that extracted keywords/sentences should align with LDA topic words to some extent. Finally, a weighted method calculates comprehensive importance to select the most important words and sentences as the post's keywords kw and key sentences ks.

3.4.1 TextRank-based Keyword/Sentence Importance Calculation This paper uses TextRank to extract key information from posts. Its basic idea is: first segment text into constituent units (words, sentences), then build a graph model, and use a voting mechanism to rank components in the text. The advantage of this algorithm is that it does not require prior learning training on multiple documents—it can complete keyword extraction and summarization based solely on single document information, with a concise and effective process.

The TextRank model structure is a directed weighted graph G = (V, E), consisting of a point set V and an edge set E, where E is a subset of V×V. The weight of the edge between any two points V_i and V_j in the directed graph is w_{ji}. For any given point V_i, IN(V_i) represents the set of points pointing to it, and OUT(V_i) represents the set of points that V_i points to. The score WS(V_i) of point V_i is defined as:

$$WS(V_i) = (1-d) + d \times \sum_{V_j \in IN(V_i)} \frac{w_{ji}}{\sum_{V_k \in OUT(V_j)} w_{jk}} WS(V_j)$$

In formula (6), d is the damping coefficient with a value range of 0 to 1, representing the probability of pointing from a specific point in the graph to any other point, with a default value of 0.85. In the above formula, WS(V_j) is the score of point V_j, which is calculated through recursive iteration, so each point's score needs to be assigned a random initial value.

The goal of keyword extraction is to automatically extract meaningful words or phrases from given text. The steps include: (1) Segment text M by sentence S_i, using k_{i,j} to represent words in sentences; (2) Build graph G = (V, E), where V is the set containing k_{i,j}, and E uses co-occurrence windows to build edges between points; (3) Iteratively calculate node weights according to formula (6) until convergence; (4) Sort points by weight in descending order; (5) Extract the top N words, and if adjacent phrases are formed, combine them into multi-word keywords.

Similarly, in the above steps, replace words with sentences to extract key sentences.

3.4.2 Word2Vec-based Topic Similarity Calculation for Keywords/Sentences The Word2Vec model is a word vector model proposed by Google that attempts to determine word meaning by analyzing a word's neighbors (also called context). By training a Word2Vec model, the distance

between word vectors can be used to represent semantic similarity between words.

This paper trains a Word2Vec model to obtain word vectors for all vocabulary in the corpus, then uses word vectors to calculate similarity between words/sentences in the text and topic words. Word similarity is calculated using cosine similarity, where a and b represent word vectors of two words:

$$\cos\theta = \frac{a \cdot b}{||a|| \cdot ||b||}$$

Calculate the similarity between each word in the document and the topic words in the document's LDA topic, taking the highest topic word similarity as the word's topic similarity:

$$\cos\theta_i = Max_{j \in Topicwords}(\cos\theta_j)$$

Then, using the post's word collection as the main body, perform normalization to obtain the word's topic similarity:

$$sim_i = \frac{count_i \cdot \cos\theta_i}{\sum_{k \in AllWords} \cos\theta_k}$$

Where count_i represents the occurrence frequency of word i, and k is a word appearing in the document.

For calculating sentence-topic similarity, this paper uses the sum of similarities of the top m words (default m=3) with the highest similarity to topic words in the sentence, normalized by the sum of similarities of all sentences in the document, as the sentence's topic similarity.

3.4.3 Weighted Calculation of Keyword/Sentence Importance This paper comprehensively uses topic similarity and TextRank importance to determine the importance of words and sentences in documents. The importance of each word in a document is calculated using formula (10):

$$I_i = w \times Sim_i + (1 - w) \times TextRank_i$$

Where w represents the weight of topic similarity with a value in [0,1], TextRank_i represents the TextRank importance of the word, and Sim_i represents the word's topic similarity. Similarly, the importance of sentences in a document can also be calculated using formula (10).

Finally, document vocabulary and sentences are sorted in descending order by weighted importance I, intercepting the top T words as keywords kw and the top N sentences as key sentences ks.

4 Experimental Results and Analysis

Automotive products are among the most complex industrial products, with a huge technical system, variable market demand, and high R&D and manufacturing costs. Additionally, automotive products have high value and are closely related to people's lives, making them important products of universal concern among various industrial products. Therefore, this paper takes automotive forums as an example to conduct experiments on topic knowledge tuple extraction in professional social media.

4.1 Automotive Text Crawling

This paper developed a Scrapy-based crawler program to capture automotive review posts from the Autohome forum, selecting 10 popular model forums including Magotan, Accord, and Camry forums. The crawling content includes post titles, main content, and image text, with a time range from September 2016 to September 2017. Posts with empty content or fewer than 5 characters were deleted, as were overly long spam posts (more than 500 characters but containing no more than 20 different characters). A total of over 100,000 automotive review posts were crawled.

4.2 Text Topic Extraction

4.2.1 Training LDA Model and Outputting Topic List This paper uses the "Topic Modeling with Latent Dirichlet Allocation" library in Python to implement the algorithm process described in Section 3.2.1. First, posts are preprocessed to remove common words, place names, brand names, and other nouns. Then model parameters are selected: number of topics K=20, Dirichlet distribution hyperparameters α=0.1, =0.01, iterations=100. After training the LDA model with preprocessed posts, the initial topic list T_0 is obtained. After obtaining T_0 through LDA, synonym merging and meaningless word removal are performed on topic words under each topic. For example, "engine" and "motor" are merged into "engine," "tire" and "tyre" are merged into "tire," etc. Then, according to the topic deduplication method in Section 3.2.2, the top W=20 words from each topic are used for similarity calculation, with a topic merging similarity threshold t=0.1, meaning topics with similarity exceeding 0.1 will be merged to obtain topic list T. After running the deduplication model, the original 20 topics in the LDA topic list are merged into 10 topics, as shown in Table 2, where the "Topic" column shows topic names manually assigned based on the top 10 topic words with highest distribution from the LDA algorithm.

Table 2. Deduplicated Topic List

Topic ID	Topic Name	Top 10 Topic Words with Highest Distribution
1	Car Purchase Price and Procedure	discount, sales, loan, price, pickup, landing, purchase tax, insurance, order, markup
2	Car Comparison and Evaluation	configuration, power, sport, fuel consumption, space, gearbox, rear row, interior, landing, safety
3	Car Modification Discussion	modification, navigation, camera, headlight, upgrade, hub, installation, radar, original car, xenon
4	Car Maintenance Discussion	engine oil, maintenance, Mobil, filter, cleaning, filter element, throttle, spark plug, antifreeze, air conditioning
5	Fault, Anomaly and Repair	abnormal noise, sound, rear-end collision, brake, jitter, problem, gearbox, jerking, stall, engine
6	Usage Help	expert, help, please, assist, come in, inform, advise, car friend, excuse me, guide
7	Car Purchase Showcase	physical store, work, pickup, moderator, certification, color, good-looking, recommendation, interior, system
8	Hub and Tire Discussion	tire, hub, tire pressure, impact, spare tire, original factory, tire repair, Michelin, positioning, wear
9	Activities and Social	guess car, activity, WeChat, car club, consultation, exchange, join, software, support, music
10	Others	delete, this floor, administrator, essence, leading, forum, automatic, post, like

4.2.2 Using LDA Model to Output Document Topics Using the trained LDA model, we run it in reverse to output the topic distribution for each post. Partial document topics are shown in Table 3 , which displays the top 2 topic IDs and topic names with highest probability.

Table 3. Topics Extracted from Posts

Post Title	Post Content	Topic 1	Topic 2
FAW-Volkswagen Magotan B7L engine design defect causes valve collision	Volkswagen Magotan purchased on August 2, 2013! Drove 48,000 km, yesterday after work while driving normally, the car suddenly shook violently, lost power and stalled, then couldn't start…	5. Fault, Anomaly and Repair	2. Car Comparison and Evaluation
First close contact with Magotan B8	Colleague wants to buy a car, insists on sitting in my car after work, wants me to take him to see cars…	2. Car Comparison and Evaluation	7. Car Purchase Showcase
B7 Magotan nearly four years, some questions for experienced drivers	~[Tire wear update] 18,000 km vehicle condition explanation	6. Usage Help	8. Tire Discussion
Others are getting new models, I'm getting the old 1.8 Comfort model	Total 12,000 posts	7. Car Purchase Showcase	4. Car Maintenance Discussion

4.3 Topic Sentiment Extraction

This paper selected 12,000 posts from the corpus, used the LDA model to extract topics for each post (belonging to the 10 topic categories above), and formed an 8-person annotation team to annotate sentiment orientation related to the text topics through division of labor. To ensure annotation quality, cross-validation was performed, i.e., each post was annotated by 2 people, and posts with different annotation results were re-annotated. The specific label distribution is shown in Table 1. Posts were divided into training and test sets at a 2:1 ratio, with 8,000 and 4,000 posts respectively. This paper simultaneously used T-LSTM, LSTM, and SVM models for sentiment analysis experiments and compared the results.

4.3.1 Manual Dataset Annotation Using the T-LSTM model for sentiment analysis requires high-quality annotated topic sentiment orientation labels that should be annotated around the text's own topics.

4.3.2 Model Hyperparameter Selection 5-fold cross-validation on the training set was used to select model hyperparameters, which were also used

in subsequent experiments. The T-LSTM and LSTM models selected the same hyperparameters: vocabulary size w range (5,000, 20,000) with search interval 1,000; word vector dimension d range (50, 200) with search interval 10; LSTM hidden nodes H_l range (100, 1,000) with search interval 100; dropout rate drpt range (0.5, 0.9) with search interval 0.1. To reduce model complexity, the MLP hidden layer number is 1, with hidden nodes H_m range (50, 200). Grid Search was used to select the optimal average accuracy group, shown in Table 4. Additionally, the SVM model regularization constant C is 1.0.

Table 4. T-LSTM Hyperparameter Values

Hyperparameter	Value
Word vector dimension	100
LSTM hidden nodes	300
MLP hidden nodes	100
Dropout rate	0.5

4.3.3 Experimental Results Analysis This experiment is a three-classification problem. In the test set containing 4,000 posts, the numbers of "positive," "negative," and "neutral" labels are 845, 743, and 2,412 respectively. Different sample sizes of training sets were used during training, with effects shown in Figure 4 [Figure 4: see original paper]. When the training set size reaches 4,000, T-LSTM begins to outperform LSTM and SVM. Due to the higher complexity of the T-LSTM model compared to the other two models, it has relative advantages when the training set is sufficiently large. According to Figure 4, when the training set size is 8,000, the accuracies of T-LSTM, LSTM, and SVM on the test set are 84.9%, 82.6%, and 80.4% respectively. T-LSTM improves accuracy by 2.3% and 4.2% compared to LSTM and SVM.

Tables 5, 6, and 7 are the confusion matrices of LSTM, SVM, and T-LSTM models on the test set, where labels "0," "1," and "2" represent "positive," "negative," and "neutral," with actual quantities of 845, 743, and 2,412 respectively. The confusion matrices clearly show the correct and incorrect prediction situations of each model. In contrast, in the T-LSTM model confusion matrix, the numbers of correctly predicted labels "0," "1," and "2" are 621, 591, and 2,161 respectively, all higher than the corresponding correctly predicted numbers in the other two models.

The analysis shows that the T-LSTM model, by incorporating topic features and improving the LSTM structure, has advantages when the sample set is sufficient, can leverage LSTM's advantages in processing sequential data, and improves sentiment analysis accuracy in the topic direction by inputting topic information into the model.

4.4 Topic Keyword and Sentence Extraction

Keyword and sentence extraction uses two posts as examples (post content see Post 1 and Post 2 in Table 8) to demonstrate the extraction process and results.

4.4.1 TextRank Importance Calculation and Word2Vec Topic Similarity Calculation **TextRank importance calculation:** Through Python programming, experiments were conducted on sample automotive post texts according to the algorithm in Section 3.4.1. Model parameters: co-occurrence window length K=6, keyword count T=20, minimum keyword occurrence frequency=1. TextRank-based keyword calculation results are shown in Table 9 , and key sentence calculation results are shown in Table 10 .

Word2Vec topic similarity calculation: This paper uses the Gensim library in Python to train the Word2Vec model, setting model training parameters: word vector dimension size=100, learning rate alpha=0.05, minimum word frequency min_{count}=3, training window size window=5. After segmenting over 120,000 posts, they are input into the model to obtain word vectors for all vocabulary. Then, the LDA topic of each post is obtained, and the topic similarity of words in the post is calculated according to the method in Section 3.4.2. For example, Post 1's topic is "5. Fault, Anomaly and Repair," with topic words including: "abnormal noise, sound, jitter, rear-end collision, brake, problem, gearbox, jerking, stall, engine." Post 2's topic is "2. Car Comparison and Evaluation," with topic words including: "configuration, power, sport, fuel consumption, space, gearbox, rear row, interior, safety." Words in Post 1 such as "jitter," "abnormal noise," "problem," "gearbox," and "engine" also appear in this post's LDA topic words, so its topic similarity is higher. The Word2Vec-based keyword topic similarity results are shown in Table 11 , and key sentence topic similarity results are shown in Table 12 .

4.4.2 Weighted Importance Calculation Based on TextRank importance and Word2Vec topic similarity, weighted importance of keywords and sentences is calculated according to formula (10), where topic similarity weight w is set to 0.5. Keyword and key sentence calculation results are shown in Tables 13 and 14 respectively.

Table 13. Weighted Keyword Importance

Post	Keywords	Weighted Importance
Post 1	problem, engine, jitter, abnormal noise, unable	0.021, 0.013, 0.013, 0.013, 0.012
Post 2	interior, landing, colleague, space, rear row	0.015, 0.009, 0.009, 0.025, 0.011

Table 14. Weighted Key Sentence Importance

Post	Key Sentences	Weighted Importance
Post 1	Drove 48,000 km, yesterday after work while driving normally, the car suddenly shook violently…	0.176
Post 1	I'm convinced, really convinced, after a few years I can basically repair cars myself.	0.115
Post 2	The 330 luxury Magotan displayed in the showroom costs less than 300,000 landing…	0.110
Post 2	The car's appearance and space are both good, giving the appearance a thumbs up…	0.069

In Table 13, Post 1's weighted keywords are "problem," "engine," "jitter," "abnormal noise," "unable." Compared with using the two methods separately, the weights of "problem" and "engine" increase, and words with high topic similarity like "jitter" and "abnormal noise" are selected as keywords. Compared with Table 8, the selected keywords are more reasonable for Post 1's content. In Post 2, after weighted calculation, words with high topic similarity like "space," "interior," and "landing" are selected as keywords, while "look" and "texture" are excluded. Compared with Table 8, the keywords are more relevant to Post 2's topic.

Key sentence extraction using Word2Vec-based topic similarity algorithm and TextRank algorithm with weighted calculation also achieves the same effect, with calculation results shown in Table 14.

4.5 Topic Knowledge Tuple Extraction Model Integration

The extraction methods for text topic t, sentiment orientation p, keywords kw, and key sentences ks are integrated. The integrated model converts the corpus into structured topic knowledge tuples for storage, facilitating retrieval and use of required knowledge in product innovation.

First, the integrated model is initialized by inputting the text corpus D into the LDA and LSTM models for training to obtain trained model files. Then, an API interface program is written to load the LDA and LSTM model files, output model results, and implement the keyword/sentence extraction process. Finally,

inputting a single text title and content outputs a structured topic knowledge tuple. This API implements the mapping from text title h, text content c, and topic list T to topic knowledge tuple, i.e., f_u: (h, c, T) → (h, c, t, p, kw, ks).

By calling this API, unstructured text can be converted into structured topic knowledge tuples. Example output results are shown in Table 8, where text topic t and key sentences ks take the Top 1 item, and keywords kw take the Top 5 items. Finally, 2,000 texts were randomly selected to call this model API for manual verification. The number of qualified topic knowledge tuples extracted was 1,382, with a qualification rate of 69.1%. A qualified topic knowledge tuple means all elements in the tuple are correctly extracted—only when text topic, sentiment orientation, keywords, and key sentences are all correctly extracted is it considered qualified.

For keyword and sentence extraction, 2,000 posts were randomly selected for manual verification. Using TextRank alone extracted 1,402 qualified keyword/sentence items, with a qualification rate of 70.1%. The weighted algorithm integrating topic similarity extracted 1,562 qualified keyword/sentence items, with a qualification rate of 78.1%, an 8% improvement.

5 Conclusion and Recommendations

This paper proposes a topic knowledge tuple extraction method for professional social media. First, the LDA model extracts topics from professional social media texts, and topics are clustered and deduplicated to form a topic list. Second, a T-LSTM model suitable for professional social media texts is constructed by integrating text topics. Then, the TextRank algorithm and topic similarity algorithm are fused to extract keywords and key sentences from texts for explaining and supplementing topics and sentiment orientation. Finally, the above models are encapsulated, and a complete topic knowledge tuple extraction scheme is formed by converting post texts into topic knowledge tuples through an encapsulation program.

The proposed model can better adapt to the text characteristics of professional social media forums. In topic extraction, it further reduces topic redundancy; in topic sentiment analysis, it performs sentiment analysis around text topics, improving sentiment orientation classification accuracy; in keyword/sentence extraction, the extracted keywords and sentences are more relevant to text topics. This paper constructs a complete and systematic extraction scheme for automotive social media topic knowledge tuples. Experimental verification shows that the accuracy of extracted topic knowledge tuples reaches 69.1%. Additionally, combining deep learning with traditional semantic analysis technology and introducing it into this topic knowledge tuple extraction scheme is an important feature of this extraction method.

Future research directions for topic knowledge tuple extraction in professional social media include: (1) Using topic words output by the LDA model to construct ontologies, establishing hierarchical structures and mapping relationships

between topics, while calculating association strength between topic elements; (2) Integrating supervised or semi-supervised deep learning methods to extract keywords and sentences, thereby improving the accuracy of topic knowledge tuple extraction.

References

[1] Wen Tingxiao, Hou Jingchuan, Gong Jiaoteng, et al. Construction of Chinese text knowledge elements and its practical significance[J]. Journal of Library Science in China, 2007, 33(6): 91-95.

[2] Bu Qu. Research on brand community network structure and member interaction content[J]. Modern Business Trade Industry, 2016, 37(4): 55-56.

[3] Wu Jing. Discussion on text construction characteristics of online forums[J]. News Research Journal, 2016, 7(4): 55-56.

[4] Wang Zhijin. Objectives and tasks of knowledge organization[J]. Information Studies: Theory & Application, 1999, 22(2): 65-68.

[5] Wen Youkui, Wen Hao, Xu Duanyi, et al. Text knowledge indexing based on knowledge elements[J]. Journal of the China Society for Scientific and Technical Information, 2006, 25(3): 282-288.

[6] Jiang Yongchang. Research on basic principles of knowledge construction (Part 2)—Technical support for knowledge construction[J]. Library and Information Service, 2009, 53(6): 100-104.

[7] Liu Miao, Wang Yu. Construction of journal literature knowledge element database based on topic sentences[J]. Journal of Intelligence, 2012(11): 145-149.

[8] Yang Liang. Research on key technologies of text sentiment analysis for social media[D]. Dalian: Dalian University of Technology, 2016.

[9] Yin Y, Song Y, Zhang M. Document-level multi-aspect sentiment classification as machine comprehension[C]//Palmer M. Proceedings of the conference on empirical methods in natural language processing. Copenhagen: Association for Computational Linguistics, 2017: 2044-2054.

[10] Blei D, Ng A, Jordan M. Latent dirichlet allocation[J]. Journal of Machine Learning Research, 2003(3): 993-1022.

[11] Tu Haili, Tang Xiaobo, Xie Li. Research on user requirement mining model based on online reviews[J]. Journal of the China Society for Scientific and Technical Information, 2015, 34(10): 1088-1097.

[12] Alex G. Long short-term memory[M]//Supervised sequence labeling with recurrent neural networks. Berlin: Springer, 2012: 1735-1780.

[13] Liang Jun, Chai Yumei, Yuan Huibin, et al. Sentiment analysis based on polarity shifting and LSTM recurrent network[J]. Journal of Chinese Information Processing, 2015, 29(5): 152-159.

[14] Mihalcea R, Tarau P. TextRank: Bringing order into texts[C]//Rille. Proceedings of the conference on empirical methods in natural language processing. Barcelona: Association for Computational Linguistics, 2004: 404-411.

[15] Han Longshi. Internet + automotive new thinking and business model innovation[J]. Enterprise Management, 2016(7): 104-106.

Author Contributions

Lin Jie: Determined the paper structure and wrote the paper.
Miao Runsheng: Designed experiments, conducted data collection and experimental analysis, and wrote the paper.
Zhang Zhenyu: Conducted data organization and revised the paper.

Note: Figure translations are in progress. See original paper for figures.