w

KeywordMap: Attention-based Visual Exploration for Keyword Analysis

Yamei Tu*
The Ohio State University

Jiayi Xu†
The Ohio State University

Han-Wei Shen‡
The Ohio State University

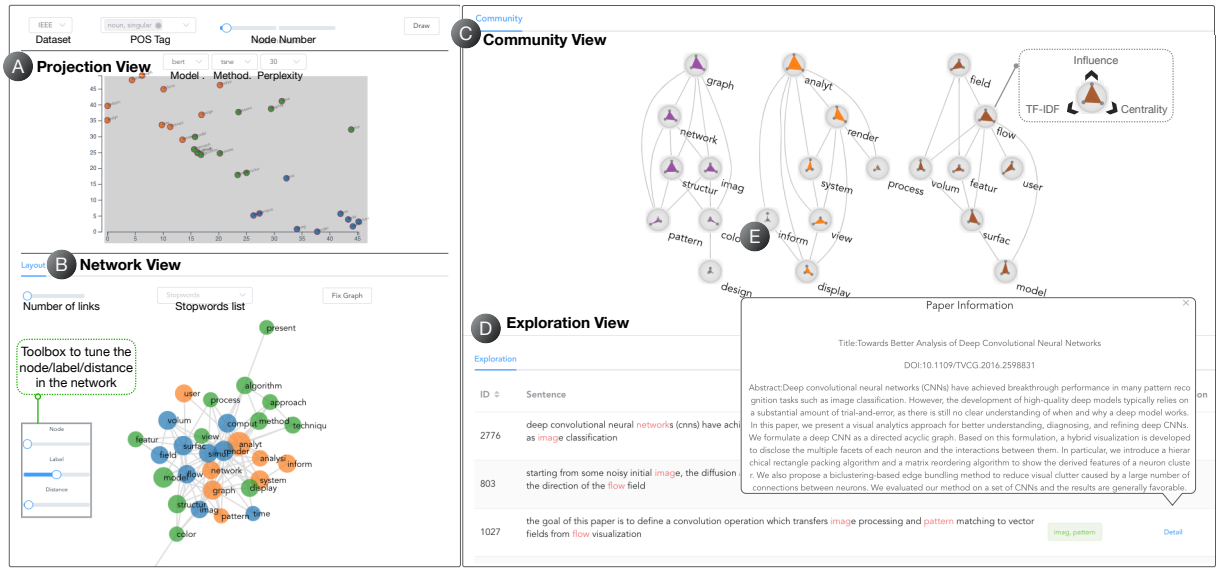

Figure 1: The KeywordMap system consists of four views: (a) Projection View: projection of node embeddings which are extracted from the fine-tuned model. (b) Network View: undirected keywords network capturing the local structures of brushed keywords in the *projection* view. (c) Community View: novel glyph-based visualization to present the keywords within each community. (d) Exploration View: retrieving related text for the target keywords or keyword relationships.

Abstract

With the high growth rate of text data, extracting meaningful information from a large corpus becomes increasingly difficult. Keyword extraction and analysis is a common approach to tackle the problem, but it is non-trivial to identify important words in the text and represent the multifaceted properties of those words effectively. Traditional topic modeling based keyword analysis algorithms require hyper-parameters which are often difficult to tune without enough prior knowledge. In addition, the relationships among the keywords are often difficult to obtain. In this paper, we utilize the attention scores extracted from Transformer-based language models to capture word relationships. We propose a domain-driven attention tuning method, guiding the attention to learn domain-specific word relationships. From the attention, we build a keyword network and propose a novel algorithm, Attention-based Word Influence (AWI), to compute how influential each word is in the network. An interactive visual analytics system, KeywordMap, is developed to support multi-level analysis of keywords and keyword relationships through coordinated views. We measure the quality of keywords captured by our AWI algorithm quantitatively. We also evaluate the usefulness and effectiveness of KeywordMap through case studies.

Index Terms: Text/Document Data—Machine Learning—Task and Requirements Analysis

*e-mail: tu.253@osu.edu
†e-mail: xu.2205@osu.edu
‡e-mail: shen.94@osu.edu

1 Introduction

Keywords are the compact representation of documents for depicting their essential ideas. Keyword analysis has been used for document categorization, summarization [20], indexing, and clustering [34] in many domains such as Information Retrieval (IR) [5, 31, 40] and Natural Language Processing (NLP). With the rapid growth of new documents each year [32], keyword-based search has become an essential means to identify relevant information since filtering through the main text of a large corpus is very time-consuming and energy-demanding. Although keywords can be used to explain essential terminologies, individual keywords alone may not provide a complete view of the document collection. For example, a paper may present a new idea that connects two keywords that were not related to each other before. The connections between interesting keywords may also inspire some new research ideas. Since keywords are often assembled to form "topics" or "concepts" in a large corpus, forming keyword networks from document collections will provide researchers with a comprehensive view of the literature.

Topic modeling can summarize a whole corpus with important keyword groups, and hence is widely used for document visualization. Examples of the techniques include Latent Semantic Analysis (LSA) [18], Probabilistic Latent Semantic Analysis (PLSA) [12], and Latent Dirichlet Allocation (LDA) [2]. While widely used, these traditional methods have several limitations. First, topic modeling algorithms identify topics by a set of keywords. However, why certain words are placed together is not always clear [27]. Second, sometimes it is difficult to fine-tune the parameters of the topic modeling algorithms to produce good results [49]. Some graph-based

methods are proposed for keyword extraction but they do not group keywords to form topics [28].

To address these challenges, we propose a new method that utilizes attention, one of the promising ideas in recent Deep Learning literature, to capture word relationships. To make full use of the models pre-trained on large corpus collections, we introduce a process called *domain-driven attention tuning* to obtain a better word relation network preserving the domain-specific knowledge. We develop a method to take the word relation network as an input to calculate the *influence* score of each word to decide its importance. Furthermore, a visual interface is designed to present the word network and perform network-based keyword analysis. The idea of the interface is analogous to the "map", which depicts the relationship between elements and also provides detailed information about each element. We perform both quantitative and qualitative evaluations to measure the effectiveness of our method. Furthermore, we conduct case studies to demonstrate the efficacy of our system. The main contributions of our work are as follows:

• We propose a method to fine-tune Transformer-based neural networks and utilize the attention maps to build word networks. We further design a new method to calculate the importance of the words when propagating the information in the network.

• We create an attention-powered keyword visual-analytics system to efficiently help users identify important words and support multiple levels of keyword analysis. With an application on literature analysis, our system guides users to locate keywords of interest which can be concepts, methods, or topics in the literature. We assist users to drill down to the original papers to find out the related information useful for literature survey or research inspiration.

2 Related Works

2.1 Neural Network Assisted Keyword Extraction

Identifying keywords from documents is not a trivial task. Traditional methods utilizing machine learning consider the task as a classification problem. They label words as keywords or non-keywords and then predict them based on syntactic and lexical features [26,47]. Most of these methods utilize statistical features of words, such as frequency, co-occurrence, and TF-IDF. However, there is much room to improve due to the lack of semantic information, and the well-annotated supervised training datasets.

The rapid developments in NLP have brought great success to the learning of the semantic information of words. Word2vec was trained on a large corpus to present each word as a meaningful vector which supports mathematical functions to compute the word relation [29]. Suleiman et al. group words into different classes based on the cosine similarity of word vectors [42]. Ulgen Ogul et al. propose another supervised keywords extraction approach based on Word2vec [33]. Later, Transformer was proposed for sequential data to solve many downstream tasks, such as translation and summerization [44]. It outperforms Word2vec by taking the context of words into account, which is known as the attention mechanism. Our method makes full use of it to capture the word relationships, so we explain the details of Transformer as preliminaries in Sect. 3 .

2.2 Keyword Analysis

With the rapid growth in the number of text documents, keyword analysis becomes an important tool to analyze the information contained in a large corpus [22]. Among many, uncovering the structure of literature or sub-field relationships, also called knowledge domain visualizations(KDVs) [24], is the most related to our work. These works differ in how to build the knowledge graph. For example, it is a common way to utilize the word co-occurrence to build the relatedness of scientific terms [16, 24, 43]. Rosvall et al. combine the random walk with information theory to do hierarchical clustering, which facilitates users to discover the multi-level structure and relationship of large knowledge graph [38]. Besides the normal document, other data formats can also be used to establish the knowledge graph, e.g. biblio-metrics, citation networks of papers [4]. In our work, we fine-tune the pre-trained transformer to learn the word relationships for each specific dataset, which makes it scalable to different kinds of literature.

2.3 Visual Exploration in Keyword Analysis

Many visualization methods are proposed to support keyword analysis. Graph visualization represents words as nodes and encodes word relationships as edges [21,41]. Trees can be used to effectively obtain an overview of large text corpora [46]. Beck et al. incorporate word cloud, bar charts, and text highlighting in a system, SurVis, to analyze and disseminate literature collections [1]. SurVis is developed specifically for scientific researchers. However, KeywordMap can be applied to multiple areas with domain-specific datasets. Jiang et al. applies a hierarchical topic modeling algorithm to extract the topics of different domains and displays the topic evolution via the Sankey diagram, combined with a word cloud and scatter plot [16]. The evolution of topics can illustrate some patterns which are useful for data analysis. However, this temporal trend is not the focus of KeywordMap. Instead, we focus on the formation of keyword relationships according to the raw document collections. Galex facilitates analysis in three progressively fine-grained levels: discipline, area, and institution levels [19]. Visual interaction also enables users to explore arbitrary periods and any sub-area of one discipline. The smallest analysis entity in this work is word-level; In KeywordMap, we design new visualization to display the different measurements of one word, allowing users to compare diverse properties of one single word. KeyVis allows author-assisted and expert-coded keyword analysis in the Visualization community [14]. Keywords set by experts and authors are more accurate but time-consuming compared to our automatic method.

3 Background

In this section, we first introduce the background on the Transformer-based model, then move on to describe the most critical part of the Transformer, namely, the multi-head self-attention mechanism.

3.1 Transformer-based Encoders

The Transformer architecture was designed to solve neural machine translation [44]. It is composed of two components: the encoding component and the decoding component. The encoding component consists of layered encoders which convert the input to embeddings through each encoder, as illustrated in Fig. 2(A). Similarly, the decoding component translates the embeddings back into another sequence layer by layer. The pre-trained Transformer-based language model only contains the encoding component, since sequence embedding from encoders can be further fine-tuned to downstream tasks. There are many available models pre-trained on large unlabelled text, such as BERT [9], Google's TransformerXL [6], OpenAI's GPT-2 [36], XLNet [48], RoBERTa [23], etc. Based on the pre-training objectives, the models can be categorized into the AR(autoregressive) model and the AE(autoencoding) model. We choose two representative models in each type in our experiment: BERT [9] and XLNet [48] to see the scalability of our method to different pre-training models.

3.2 Multi-head Self-attention Mechanism

The structure of the Transformer is illustrated in Fig. 2(A), it is a stack of 12 encoders that has a self-attention layer and a feed-forward layer. The self-attention layer generates the contextual word embeddings as a function of all context words, known as *multi-head self-attention mechanism*. The computation is processed in three steps.

First, the input sentence with n words are converted to n word embeddings. Then the model calculate three vectors for each input

word embedding w_i: a query $\mathbf{q}_i$ and a key-value pair ($\mathbf{k}_i$, $\mathbf{v}_i$). They are created by multiplying the word embedding by three learnable weight matrices $\mathbf{W}^Q$,$\mathbf{W}^K$,$\mathbf{W}^V$. Second, to calculate the self-attention weight for the word w_i, it is trained to decide how much attention to be placed on all the other input words. The scores are computed by taking the dot product of $\mathbf{q}_i$ with keys of all other words($\mathbf{k}_1$,$\mathbf{k}_2$...$\mathbf{k}_n$). Next, the score is divided by $\sqrt{d_k}$ where d_k is the dimensionality of the key vector, and goes through the softmax operation. Now we have n softmax values indicating how much attention the word w_i pays to all words including itself. Last, to filter irrelevant words with low softmax scores, we multiply each softmax value with the corresponding $\mathbf{v}_i$ and sum all the vectors. The final vector is the embedding for the word w_i considering the influence of all the context words. In practice, we compute this as matrix operations for high efficiency. We combine the three vectors for each input token respectively to get three matrices, denoted as **Q, K, V**, where the ith row are $\mathbf{q}_i$, $\mathbf{k}_i$, $\mathbf{v}_i$.

There are many heads in one layer, each corresponding to one set of ($\mathbf{W}^Q$,$\mathbf{W}^K$,$\mathbf{W}^V$). Different heads look at different patterns existing in the sentence. We use the base version pre-trained models in our experiment, thus the number of heads in each layer is 12, the attention maps are defined as follows for each head i:

$$AM_i = softmax(\frac{Q_i K_i^T}{\sqrt{d_k}})(i = 1,2,...,12) \quad (1)$$

The n×n matrix indicates how each word interacts with all the other words within the sentence, where each row r_i implies how w_i hands out its attention to all the other words and column c_j illustrates how w_j receives attention from others.

4 Attention-Assisted Keyword Analysis Approach

Keyword analysis for a large corpus is still a challenging task for three reasons. First, due to the delicacy of human language, treating documents as a bag of words, a common technique used in the literature may lose the context which plays an important role in determining the meaning of the text. Second, a set of correlated keywords usually represents a concept or topic, but finding the correlated words to form a meaningful group is not an easy task. Lastly, traditional topic modeling algorithms often require the user to specify the number of topics in the text. However, we do not have ground truth for it in most cases. In this work, we propose a systematic method to solve these challenges, including three major components as illustrated in Fig. 2.

1. Domain-driven attention tuning: To solve the first problem, we attempt to find the relationships among words instead of treating them independently. We propose to utilize the attention mechanism to model the word relationships in the Transformer-based language model. There are many pre-trained models available to use, but the learnable weights are not trained to capture domain-specific knowledge. So we perform *domain-driven attention tuning* through supervised classification tasks, forcing the model to learn better domain-specific word dependencies, described in Sect. 4.1.1.

2. Keyword extraction: We extract the attention maps from the fine-tuned model to capture the pairwise word interactions within each sentence. Then we merge the word-to-word relationships extracted from each sentence to form a consolidated network, introduced in Sect. 4.1.2. In order to fulfill keyword extraction, we propose an algorithm to rank the keywords considering how words propagate their attention in the network, described in Sect. 4.2.

3. Interactive system: Analyzing the word network in an efficient way can mitigate both the second and third challenges. Performing community detection in the word network can capture the correlated keywords, which treats the number of communities as a dynamic value adapting to the user-defined network. The KeywordMap system is developed in a hierarchical manner to provide multi-level analysis and visual interactions. We describe how to perform analysis through our KeywordMap in Sect. 4.3.

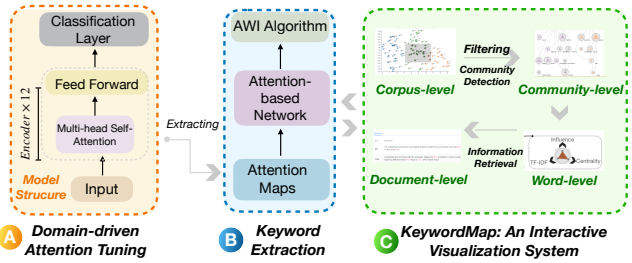

Figure 2: Pipeline of the KeywordMap.

4.1 Building Attention-based Word Network

4.1.1 Domain-Driven Attention Tuning

We propose to utilize the attention mechanism to model word relationships. Since the relationships of words may change across different domains, fine-tuning the attention to capture the domain-specific knowledge is necessary. We define this methodology as *domain-driven attention tuning*. To achieve this, the pre-trained models can be fine-tuned by adding one additional layer with specific NLP tasks, which is a typical way to perform transfer learning in the NLP area. As shown in Fig. 2(A), our model is composed of two modules: a pre-trained Transformer-based language model + a classification layer. As we feed a text sequence, the pre-trained model generates sentence embedding and token embeddings. The classification layer computes the probability of sentence embedding belonging to each of the preset labels. We use the cross-entropy as the loss function, defined as follows:

$$H(p,q) = -\sum_x p(x) log q(x). \quad (2)$$

where $p(x)$ means the predicted probability, and $q(x)$ is the actual probability. We back-propagate the loss to update the learnable parameters in both the Transformer and the classification layer. Through this supervised training, the embeddings are steered towards the correct label class. These embeddings are computed based on the intermediate result, i.e.attention maps. Therefore, it should also be improved to better capture the relationships of words as the classification performance increases.

This methodology not only provides a new perspective to capture word relationships but also makes good use of data labels. As the amount of accumulated text data is increasing at an astonishing rate, creating a well-labeled text corpus requires lots of human labor. However, text data with general topics labeled are much more available. For example, industries gather user feedback of products into general categories, but labeling them with specific consumer focuses is time-consuming and costly. Academic paper datasets often have their publication venues but lack specific topics. In other words, our domain-driven attention tuning provides a novel perspective to learn more specific word relationships by learning general labels through supervised training.

4.1.2 Constructing Word Network

As described in Sect. 3, the attention map of one sentence with n tokens is an n×n matrix. Each row r_i indicates how the word w_i allocates attention to all the n words including itself. We average the matrix across multiple heads for one sentence to include all the existing patterns and denote it as AM, so the sum of attention score in each row equals 1. Since there are 12 layers in the model, we have two different parameter settings: attention scores averaged from all layers and only the last layer. Furthermore, we also evaluate the difference raised by the layer in Sect. 5.5. After we compute the averaged AM for each sentence, we need to merge AMs from different sentences in the corpus to construct a word network.

To fulfill this, we keep incorporating the individual matrix AM^k (k=1,2...T, T is the total number of sentences) into one network,

Algorithm 1: Constructing ASNetwork from attention maps

Input: $AM^1, AM^2, ..., AM^T$
Output: *ASNetwork*
1 c_{ij}: each cell in matrix, denoting the attention score sending from w_i to w_j, w_i: stemmed word i
2 ASNetwork = (V,E), $e_{ij} \in$ E, $w_i \in V$
3 Define edge attribute sets = {sum, frequency, weight}
4 **for** *k=1,2,...,T* **do**
5 **for** c_{ij} *in* AM^k **do**
6 **if** $e_{ij} \in E$ **then**
7 e_{ij}.sum += c_{ij}
8 e_{ij}.frequency +=1
  else
9 e_{ij}.sum $\leftarrow c_{ij}$
10 e_{ij}.frequency $\leftarrow 1$
  end
  end
end
11 **for** $w_i, w_j \in E$ **do**
12 e_{ij}.weight $\leftarrow e_{ij}$.sum / e_{ij}.frequency
13 e_{ji}.weight $\leftarrow e_{ji}$.sum / e_{ji}.frequency
14 e_{ij}.weight, e_{ji}.weight $\leftarrow$ (e_{ij}.weight+e_{ji}.weight)/2;
end

defined as Attention Score Network(*ASNetwork*) following the algorithm 1. To reduce the number of nodes in the network, we use word stem when we constructing the network. Given the fact that a high-frequency relationship does not guarantee a high attention score, and a strong relationship does not promise frequent appearance, to eliminate the frequency effect in the attention score, we calculate the average attention score of each word pair(c_{ij}) as the edge weight from w_i to w_j (line 4-13). We assign three attributes to each edge; *frequency* indicates how many times the word pair appears; *sum* represents the sum of all c_{ij} existing in all AMs; *weight* is the edge weight, computed based on the other two attributes (line 12-13). As a result, e_{ij} is always in the range of [0,1]. Furthermore, a strong relationship should be defined as bi-directional attractions, so we average the bi-directional edge weights to avoid biased bi-directional edge weights (*line 14*).

4.2 Attention-based Word Influence Algorithm

After we construct the word network based on the attention maps, we need to determine the importance of words in the network. In this section, we propose a new algorithm inspired by the social network literature to compute how influential each node in the graph is. In social networks, users influence the community by posting their ideas, opinions, or retweeting others' tweets to propagate information. To study the dynamics of a network, deciding which user is more influential has been an active research topic. Romero et al. proposed a model to take the followers' status into consideration [37]. In their method, it was argued that judging only by whether one has a large number of followers can not necessarily guarantee a high influence. It should also consider whether one's followers are influential.

We find the analogy between the social network and the ASNetwork. In this study, we aim to detect the influential word in the ASNetwork, same as the influencer in social networks. For edge e_{ij} in ASNetwork, we define the word w_i as source word and w_j as target word. Many source words attending to the target word can not guarantee it is influential. It also depends on whether these source words are influential and how much attention is allocated. Therefore, influential words are defined as those who receive lots of high attention scores from other words who also receive high attention. Based on this idea, we measure the importance of words by calculating the influence score as described in algorithm 2.

Algorithm 2: Attention-based Word Influence

Input: ASNetwork
Output: Influence I_i for each word $v_i \in$ V.
1 $e_{k \to i}$: the edge weight, denoting how much attention is transitioned from $word_k$ to $word_i$.
2 **repeat**
  for $v_i \in V$ **do**
  $I_i = \sum_{(k \to i \in G)} e_{k \to i} \times I_k$
  $\text{softmax}(I_i) = \frac{e^{I_i}}{\sum_{j=1}^{N} e^{I_j}}$
  end
until I_i *converges*;

Influence score: this property indicates how influential a word propagates information to the whole network. A word's *influence* score is based on:

- the number of source words;
- how much attention each source word gives to it;
- its source words' influence scores.

We keep updating *influence* score I_i for word *i* interactively in the whole network until they are converged. According to the network analysis in social network [37], it will converge within 10 times even for large network graphs.

4.3 KeywordMap: An Interactive Attention-driven Visualization System for Keyword Analysis

Powered by our keyword analysis methods, a visual interactive system is developed to facilitate visual exploration.

4.3.1 System Requirements

We develop an interface to support keyword analysis for large corpus data. The design was done through an iterative process and the requirements listed below are defined based on (1) specific needs from experts' feedback in Information Retrieval literature. (2) general needs for literature analysis.

R1(corpus-level) Identify comprehensive sets of keywords: Constructing an overview of keywords is useful in multiple scenarios. For example, the decision-makers of a conference need to list a comprehensive set of keywords to attract diverse submissions but in the meantime maintain proper emphases.

R2(group-level) Reveal relationships among keywords: Analyzing keyword relationships often leads to the identification of interesting concepts. The essence of research in general is to solve a problem with new and better methods. Therefore the emergence of novel research ideas can be expressed by newly established links among keywords to represent new algorithms, methods, problems, and techniques.

R3(word-level) Perform multifaceted analysis of keywords: Compare different properties of keywords can reveal their importance from multiple perspectives. Some keywords are important to represent the general concepts while others are specific to particular domain knowledge.

R4(document-level) Information retrieval: Keywords set is a descriptive short representation for documents. Meaningful keywords should be able to identify specific documents for information retrieval purpose.

4.3.2 Design Tasks

Driven by the requirements, we specify the following visual design tasks and explain how they should be accomplished by our visualization system.

T1 Display an overview of keywords The system should provide an overview of all the extracted keywords. The distance between the keywords in the projection space should reflect their semantic relation, i.e., correlated words are closer to each other. Furthermore, the overview should display meaningful word clusters

to reveal the existence of keyword groups. This will assist users to brush the group of interest for further analysis.

T2 Analysis different features Each term has both syntactic and semantic features to represent unique information. Providing a multi-faceted view of the features that allow a better understanding of the individual keywords.

T3 Reveal the structure of keywords network Can the network capture both global and local structures? Global structure is useful for users to group keywords into clusters and performs further community analysis. Local structures can keep relevant words close to each other, which assures the relationships of words are meaningful in semantic space.

T4 Analyze keywords relationships Traditional topic modeling algorithms only group a set of keywords with probabilities without explaining the keyword relations. To enable more comprehensive exploration, our system facilitates users to look at the keyword relationships and understand the existence of them by tracing back to the original documents.

T6 Keep human in the analysis loop. Interpreting the keywords generated by algorithms should always incorporate human input. Human language is complex, hence it is impractical to accomplish keyword identification solely by automatic methods. Our system allows users to add words into the stopword list which will automatically update the keyword network and the following results.

4.3.3 Visual Components in KeywordMap

Motivated by the aforementioned tasks, our interface contains four coordinated views to support the visual exploration of keyword analysis. In this section, we describe how the different visual components work coordinately to serve the purpose.

Projection view: Following **T1**, we design this view to present the global structure of keyword embeddings from the fine-tuned models. The models can be either BERT or XLNet, which can be selected from the first ❯ . Given a word, it has different contextual embeddings from different sentences. We average all contextual embeddings to obtain the high-dimensional vector for each keyword. Users can click the second ❯ to select the dimensionality reduction methods: t-SNE [11] or UMAP [25] to project the vectors to 2D space, as shown in Fig. 1(A). Then, we apply the k-means algorithm to the projection space, where k is equal to the number of classes in the fine-tuning process. We color the nodes based on the k-means result to indicate the distance-based clusters. Plus, users are able to decide how many top-ranked keywords to show by dragging the slider ⚌, labeled as *Node Number* at the top of Fig. 1. Inspired by the prior work, meaningful keyword groups are built on top of a specific part of speech [8, 10]. Therefore, users are allowed to select the type of Part-Of-Speech from ❯, labeled as *POS Tag* in Fig. 1. The brush interaction is allowed to select a set of keywords for further analysis, which will trigger the *Network* component automatically.

Network view: In Fig. 1(B), we present the networks of keywords in a node-link graph which can better encode local structures to satisfy **T4**. Each node indicates a keyword, and the size of the node implies the *influence* score. Since the bi-directional edge weights are averaged when constructing the network, we draw an undirected network to reduce visual clutter. We also encode the edge weight by the distance, larger weight indicating the shorter distance between two nodes. In addition, we allow users to remove unimportant links by limiting the number of top-links through the slider ⚌ and add user-defined stop words by clicking the nodes.

Community view: According to **T3**, it is useful to identify closely correlated keywords for the purpose of understanding subareas or sub-topics. The Louvain algorithm [3] is a widely used optimization algorithm to extract communities from large networks. We apply it to the pruned network in the *network* view to perform community detection, which will give us groups of keywords.

According to **T2**, we compute three different properties for each word:(1) the averaged TF-IDF score containing the statistical information; (2) the betweenness centrality measuring the popularity in the word's network, as used in [7];(3) the influence score computed using the AWI algorithm. In order to allow comparison across different value ranges for each property, we sort each property in ascending order and map the numerical values to an index position that reflects the rank. In the glyph-based visualization, each property is mapped to a fixed axis in the radar chart. We design the novel glyph-based visualization to map each property to a fixed axis in the radar chart to allow a juxtaposed comparison. Three axes form a triangle, the area of which measures the importance of a keyword which is also a preattentive visual channel to allow instant comparison. As presented in Fig. 1(C), it is quite clear and efficient to compare three dimensions for one node and also compare across all nodes through the area of triangles.

Table 1: Epochs and performance of models trained in this paper.

Dataset	Model	Epochs	Valid. Acc.
VIS	BERT	4	0.78
VIS	XLNet	4	0.74
NEWS	BERT	2	0.88
NEWS	XLNet	1	0.84
ARXIV	BERT	2	0.91
ARXIV	XLNet	2	0.90

Exploration view: KeywordMap works as a "map" to help users navigate through a keyword network and also provide detailed information to find interesting keywords or phrases. The *exploration* view is where we start the journey by exploring the texts and sentences in the original paper, according to **T5**. After users select the keywords of interest in the *community* view, all the sentences containing the target keywords are retrieved and used to update the *exploration* view accordingly (Fig. 1(D)).

We present the retrieved sentences with highlighted keywords in a table format and group them based on the type of keyword relationships. Furthermore, we provide access to the raw paper information by clicking the *Detail* button, as shown in Fig. 1(E).

5 Evaluation

The evaluation presented in this section contains several parts. To begin with, we describe tuning processes and results for three datasets as the basis for the following evaluations. In order to prove the significance of *domain-driven attention tuning*, we evaluate how it improves the output token embeddings and intermediate attention maps. Then we compare the tuned Attention Score Network(ASNetwork) with the common co-occurrence network to demonstrate the effectiveness of the learned attention score. ASNetwork is then fed into the AWI algorithm as inputs, we evaluate its performance with the TextRank keyword extraction algorithm. Further, we also discuss the comparison of different parameter settings of the AWI algorithm.

5.1 Attention Tuning Details

We performed the attention tuning process for three datasets with two pre-trained models. The Vispubdata has publications from 1990 to 2018 with user-provided keywords, abstracts, titles etc [13]. The News dataset contains 200k news headlines from 2012 to 2018 in 41 different fields [30], and we only use 6 fields in our experiments. For the third dataset, we pre-selected publications under three subcategories of Computer Science from Arxiv and sampled 3k$\times$3 sentences for training. We trained three models for each dataset with the hyperparameters suggested in the original paper [9]. The learning rate was $2e-5$, batch-size was 32, and the number of epochs and performance can be seen from Table 1.

5.2 Evaluating the Domain-Driven Attention Tuning

Through the tuning process, learnable parameters are updated by backpropagating the loss, which can be reflected through the inter-

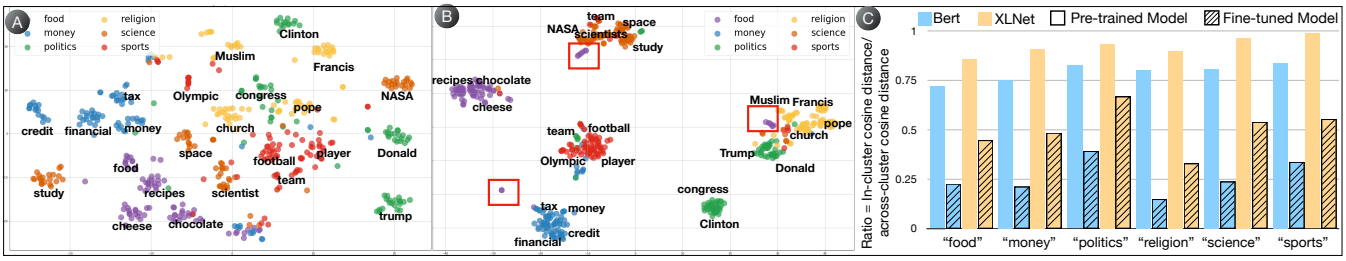

Figure 3: The projections of word embeddings from (A) pre-trained BERT model and (B) fine-tuned BERT model. (C) The ratio of pairwise in-cluster distance to the pairwise across-cluster distance for each topic.

mediate results(attention maps) and the output embeddings. In order to show the improvements, we compare how the word embeddings and attention maps are before and after the tuning process. For the same dataset, we compare two models: The Encoder only model and the Encoder+Classifier model.

5.2.1 Clusters of Word Embeddings

We conduct two evaluations to confirm that the formation of clusters takes place after the domain-driven attention tuning process. First, we perform a qualitative evaluation to project the word embeddings in 2d space in an intuitive manner. Additionally, we compute the distance change in high-dimensional space in a quantitative manner to confirm the clustering effect.

We perform the evaluation using the News Dataset because there is a clear separation among the different topic groups. We select 6 topics out of 41: Food&Drinks, Money, Politics, Religion, Science, and Sports, the color encoding is shown in Fig. 3. For each topic, we choose 4 words with high frequency and sample 20 sentences for each word. Therefore, we have 4×20 dots for each topic in the visualization, where each dot represents a word embedding.

Qualitative evaluation in projection space Fig. 3(A) is the result from the pre-trained BERT, while Fig. 3(B) is generated from the BERT+Classification model. From Fig. 3(A), we can see that relevant words are closer to each other, but they do not form a clear cluster. For example, "football", "team", "player" are neighbors due to the relevance to sports, but the distance among them is similar to that with other irrelevant words. Furthermore, some words belonging to the same group are far away from each other, such as "study", "scientists", "space", "NASA". After we tune the parameters in the Transformer by adding a classification layer, a big difference in the word embeddings can be observed as shown in Fig. 3(B). First, there is a clear boundary among different groups. Also, the number of clusters is the same as the number of labels defined in the classification task and each cluster is corresponding to one class. Since we pre-select 4 words for each topic as the ground truth, we can see the classifications are correct in most cases. For example, the topic of "Money"(●) includes "tax", "financial", "credit", "money". However, we can also identify a few outliers, e.g. "food" should be in the topic of "Food&Drinks"(●), but it appears in many other groups highlighted in red box in Fig. 3(B).

Quantitative evaluation in high-dimensional space The previous section allows us to visualize how the domain-driven attention tuning process assists in the cluster formation in 2D space. In this section, we perform quantitative measurement of the clusters in high dimensional space. As discussed before, 6 topics were chosen in our experiment, where each topic(cluster) has 80(4×20) word vectors. We compute the centroid of vectors inside/outside of each cluster j as In_j, Out_j. We calculate the ratio of pairwise in-cluster distance to pairwise across-cluster distance for each topic as follows:

$$R_j = \frac{\sum_{v_i \in C_j} cos(v_i, In_j)}{\sum_{v_i \in C_j} cos(v_i, Out_j)} (j = 1, 2..6) \quad (3)$$

For each dot v_i in the cluster C_j, we compute its distance to the two centroids: In_j and Out_j, which is also defined as in-cluster distance and across-cluster distance. Then we sum the two distances separately for all the dots within the same cluster and compute the ratio as relative distance for one cluster. The ratio should be equal or smaller than 1, smaller value means greater difference between in-cluster distance and across-cluster distance, indicating more clear boundaries. As shown in Fig. 3(C), the ratio decreases after the fine tuning which can be explained by the formation of clusters. Additionally, we can see that the distance of BERT is smaller than that of XLNet. This finding was also confirmed in 2D space, that is, BERT can generate more clear clusters than XLNet, but we only show the embedding results from BERT in Fig. 3(A)(B) to save space. The reason behind it we leave it for future work. In conclusion, we demonstrate that the distance of relevant words' embeddings has been decreased significantly due to the attention tuning process, both in projection space and high dimensional space. We conclude that the resulting word embeddings indeed capture domain-specific knowledge, as they move towards the center of the cluster they belong to after the tuning process.

5.2.2 Pattern Change of Attention Maps

Previously, we show how the *domain-driven attention tuning* injects knowledge to the word embeddings. However, how the attention map changes as a result of the embedding improvements is still not clear. In this section, we analyze how the attention map changes and whether the changes indicate domain knowledge learning.

Shown in Fig. 4(A), we randomly pick two sample sentences and compute the averaged attention maps from multiple heads in the last layer of the Transformer. We visualize the attention maps as heatmaps to provide a direct recognition of how the attention pattern changes through *domain-driven attention tuning*. In the heatmap, each row represents how each word allocates attention to all the others, while each column denotes how much attention it receives from all the others. Darker color stands for greater values.

In Fig. 4(A), a1&a2 present the attention maps from the BERT while a1'&a2' show the results from the BERT+Classification model. It is clear that in the BERT, each token gives high attention weights to itself, known as *diagonal* pattern, which is one of the frequently appearing patterns of attention [17]. While in the BERT+Classification model, each token will not only focus on itself, but also refer to other relevant tokens. This pattern is referred to the *heterogeneous* pattern, which is more likely to capture interpretable linguistic information for language understanding [17]. For example, "machine" and "learning" will give high attention to each other in the first sentence. In the second sentence, some high-weight groups pop up by giving high attention to each other. Plus, some important keywords receive higher attention from the context words, such as "particles", denoting its importance in this sentence.

The above result shows the attention matrix from one sentence. However, in our method, we combine individual attention matrices into a consolidated network. Below we explain how to show the pattern changes of the large network. As we know, a directed network G=(V,E) can be converted to a $|V| \times |V|$ adjacency matrix c, and cell c_{ij}=0 if there is no corresponding edge in G. Since the graph can be quite large, we cannot directly show the entire matrix including all

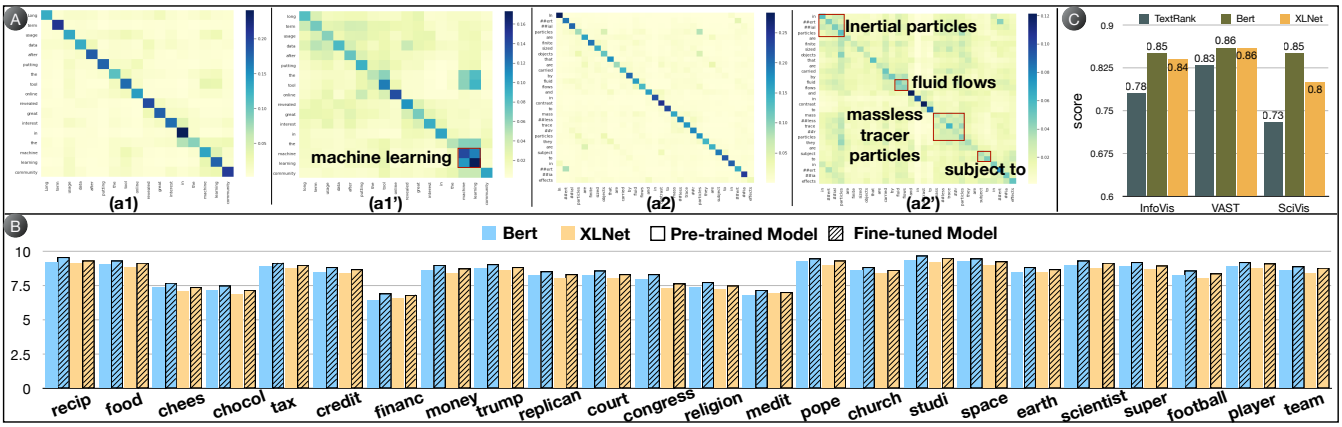

Figure 4: (A) The averaged attention map for 12 heads in the last layer of two sample sentences. Sentence1: *Long-term usage data after putting the tool online revealed great interest in the machine learning community.* Sentence2: *Inertial particles are finite-sized objects that are carried by fluid flows and in contrast to massless tracer particles they are subject to inertial effects.* (a1) Sentence1+pre-trained; (a1') Sentence1+fine-tuned; (a2) Sentence2+pre-trained; (a2') Sentence2+fine-tuned; (B) The information entropy of how each word sends out attention to all the others in different models. (C) Normalized Discounted Cumulative Gain(NDCG) scores of the AWI and the TextRank for three sub categories of Vispubdata.

the tokens. Nevertheless, we are able to calculate the information entropy of each row in the matrix, which measures the uncertainty of the attention weights sent out by a token. We expect the entropy would be higher in the BERT+Classification Model. The reason is that each token will attend more to others being injected with the domain knowledge instead of always giving high attention to itself. In this way, it would be harder for us to predict given a token, which other tokens it will attend to with high weights.

To compute the entropy, first, we convert the consolidated AS-Network to the matrix, then normalize the sum of outgoing attention to 1 in each row. Then, the matrix value in cell c_{ij} is considered as the probability of $word_i$ attends to $word_j$, defined as $P(j|i)$. We perform this for the four different model settings. We assign color channel to model and texture channel to model status, as shown in the legend of Fig. 4(B). Then, we calculate the information entropy for each $word_i$ in the final sparse matrix as follows:

$$H(i) = -\sum_{j=1}^{|V|} P(j|i) log P(j|i) \tag{4}$$

As shown in Fig. 4(B), the length of the bar indicates the entropy. Compared with two pre-trained models, both fine-tuned models have higher entropy values, which also confirms our hypothesis. We believe that the *domain-driven attention tuning* process can change the pattern of attention maps from *diagonal* pattern to *heterogeneous* which is more likely to capture semantic meaning.

5.3 Evaluating the ASNetwork

ASNetwork is generated based on the attention matrix to model the relationships between words, which is also the input to our AWI algorithm. The co-occurrence matrix is one of the most popular methods to capture word relationships. It is constructed by computing how many times the two words are co-occurring in a sliding window. In order to compare the attention matrix with the co-occurrence matrix, we generate two attention matrices(BERT-based/XLNet-based), two co-occurrence matrices(window-size=2/5). We convert four matrices to an ASNetworks and feed them into the AWI algorithm to compare the generated keywords rankings.

The result is shown in Table 2, (row1-row4) are four keyword rankings related to this evaluation. There is no big difference within top-3 keywords. However, starting from the rank4, the attention matrix can retrieve some informative keywords. We highlight these keywords in red, they are specific to provide insight about subcategories in visualization literature. For example, *graph, tree* are hot research topics in information visualization, *surface* is more likely related to scientific visualization, like stream surface. We conclude that the co-occurrence matrix grabs general concepts, while the attention matrix outperforms it by catching the domain-specific keywords due to the *domain-driven attention tuning*.

We also use popular methods to compute the similarity between words, e.g. WordNet, but the result is not satisfactory. We believe it is due to the following reasons. First, the word relationship is not equal to semantic similarity. Second, the pairwise relationships should be dynamic according to different datasets.

5.4 Evaluating the AWI Algorithm

We propose the AWI algorithm to rank words in the network based on information propagation. In this section, we evaluate the effectiveness of it through a comparison with TextRank. TextRank [28] is a popular method based on Google's PageRank algorithm [35], it is widely used in text summarization and keyword extraction.

We use the Vispubdata since author-defined keywords are provided in the raw dataset which can be used as the ground truth. We divide the Vispubdata into three categories: InfoVis, SciVis, and VAST. Applying TextRank and AWI(BERT&XLNet) to the three sub-datasets gives us nine keyword rankings in total. We define a linear function to assign a relevance score to each top-K word in the ground truth ranking: $rel_i = K - i$, where i is the index of the word w_i. Therefore, the word that appears first(smaller i) in the ground truth ranking has a higher relevance score.

Evaluating the rankings is a basic task in information retrieval. There are three most popular measurements to quantify the ranking performance: MRR(Mean Reciprocal Rank) [45], MAP(Mean Average Precision) [39], NDCG(Normalized Discounted Cumulative Gain) [15]. MRR and MAP are commonly used as binary relevance based matrices, meaning output is either relevant or irrelevant. This is not applicable in our case because we do not have the absolute list for keyword and non-keyword. Therefore, we choose to use NDCG to evaluate the ranking order. The optimal ranking is that word with high relevance score appears first. This property can be evaluated:

$$DCG_K = \sum_{i=1}^{K} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{5}$$

We set K=30 here. The DCG score will be high if the top word in the ranking(smaller i) is proved to be important in the ground truth ranking(higher rel_i value).

$$NCDG_K = \frac{DCG_K}{IDCG_K} \tag{6}$$

NCDG is computed as the ratio of the DCG to IDCG, where

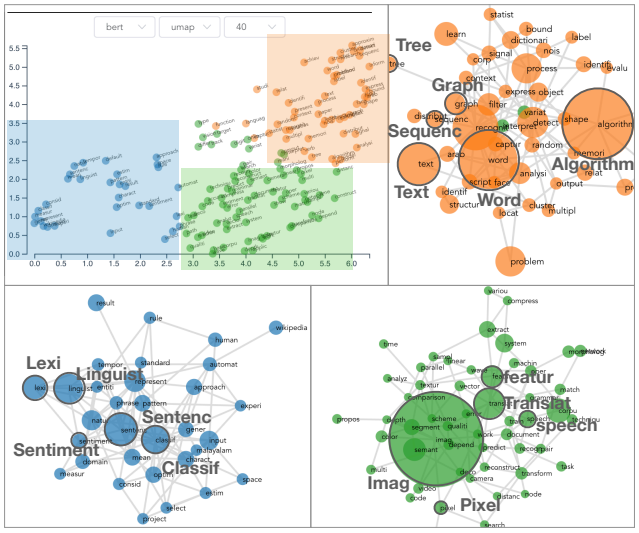

Figure 5: Brushing each clustering to see the local keyword structure in the *network* view.

IDCG(Ideal Discounted Cumulative Gain) is applying DCG formula Equation 5 to ground truth ranking. NDCG is proposed to normalize the value to [0,1], 1 means perfect ranking that is the same as ground truth. We compute the NCDG scores for nine result rankings and visualize them in a grouped bar chart Fig. 4(C). Compared with TextRank, our method has a higher NCDG score in all three categories. However, the difference in the InfoVis and VAST is larger then VAST, we infer that these two areas have a greater need to learn the domain knowledge.

5.5 Evaluating Parameter Settings

In our method, there are two places requiring hyper-parameters. The first one is choosing which pre-trained language model to use: BERT or XLNet. For the pre-trained model, there are 12 identical encoders stacked in the Transformer. The second parameter is whether to extract the attention map from the last encoder or averaging over all encoders is another parameter we evaluate.

We extract the top10 keywords with different parameter settings and present them in Table 2(row3-row6). The row3-row4 shows the ranking results computed from the last layer attention maps while row5-row6 presents the results averaged from all layers. We conclude that *all layers* is more likely to extract some general keywords, such as "model", "system", "algorithm". While *last layer* can extract some specific domain-related keywords, such as "network", "graph". Since the last two layers encode more task-specific features that improve the classification task, while earlier layers can capture more fundamental features [17], we decide to use the last-layer attention maps to construct the ASNetwork since this can generate better results than averaging over all layers. The top10 keywords generated by XLNet are more concrete than those from BERT. However, both methods can generate top-10 rankings including domain-specific keywords, which proves the scalability of our method.

6 Case Studies

We conduct case studys for real-world datasets to demonstrate the usefulness of KeywordMap and ask domain experts to verify the results and insights. All the names mentioned in our case study are pseudonyms to comply with anonymity and privacy.

We use two datasets in the case studies: Vispubdata and Arxiv with details shown in Sect. 5.1. Before the case study, we conducted a 15-20 minutes tutorial section to describe the KeywordMap system. Each case study is aimed for different target users, and all the backend pipeline follows the same method introduced in Sect. 4.

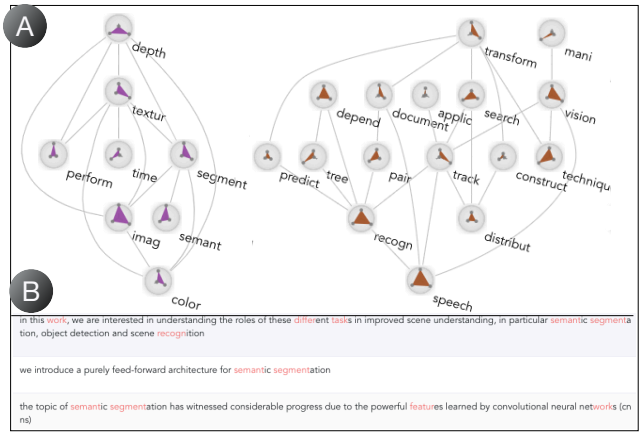

Figure 6: (A)Representative communities in the computer vision area. (B)Retrieved text related to semantic segmentation.

6.1 Case Study 1: Exploration of VIS Literature

We study Vispubdata containing all the IEEE VIS publications from 1990 to 2018. Our user David is a graduate student who has strong interests in Visualization. He wants to enhance his knowledge of visualization and identify interesting ideas to start his research.

To obtain a high-level view of the Visualization literature, David selects the Vispubdata and only displays the top 30 noun keywords. He tunes the projection through different parameters and finds that "model=BERT, Method=t-SNE, hyper-parameter=30" can produce clear boundaries among three sub-groups in the Visualization area. He brushes a large area including all the keywords, which automatically updates the *network* view with a keyword network. He filters unimportant links and keeps the most critical top-100 links. From the Fig. 1(B), it is clear to see that there are some general keywords linked to each other(●), such as "present, algorithm, method, approach, technique, process". Similar words are close to each other or connected by links, which can be evidence of meaningful local structure preserved in the network. Then he clicks on the "Fix Graph" button to generate communities from the network.

There are three sub-groups shown in different colors, where each word is clearly presented in the novel glyph-based visualization, as shown in Fig. 1(C). He decides to start the exploration with some important concepts to gain an overview of the literature. In the *community* view, he identifies that "flow" is a prominent keyword in the third community(●). He identifies that it shares many connections with other keywords, such as "model", "surfac(surface)", "volum(volume)", "user", "field", "featur(feature)". This is a new topic for David, then he confirms that this is a hot topic by checking the retrieved documents. Similarly, he confirms that "graph" is another popular keywords(●), it is related to information visualization, such as "structure, network". He believes this glyph-based design provides an efficient way to identify the important concept.

Later, David explores whether there exist interesting keyword relationships to inspire new research ideas. The relation between "imag"(image) and "pattern" attracts his attention. He goes through the related document and finds one document explaining why Convolutional Neural Network has good performance in pattern recognition through visualization, shown in Fig. 1(D). David believes the document retrieved is necessary to grasp the idea efficiently.

6.2 Case Study 2: Exploration of Computer Vision

Arxiv dataset includes papers from three sub-areas in Computer Science: DS(data structure and algorithm), CL(computation and language), CV(computer vision). We invited a researcher in computer vision(CV), Selina, who intended to identify important keywords and interesting papers related to CV through our KeywordMap interface.

To begin with general concepts, she moves the slider ⚌ to select

Table 2: Top10 Keywords with Different Methods

Method	Rank#1	Rank#2	Rank#3	Rank#4	Rank#5	Rank#6	Rank#7	Rank#8	Rank#9	Rank#10
Co-occurrence(window=2)	visual	data	techniqu	interact	user	present	method	system	analysi	approach
Co-occurrence(window=5)	visual	data	techniqu	interact	user	method	present	system	analysi	model
ASNetwork(XLNet+last layer)	visual	data	system	inform	network	graph	cluster	analysi	user	tree
ASNetwork(BERT+last layer)	data	visual	analyt	model	render	surfac	graph	system	comput	volum
ASNetwork(XLNet+all layers)	data	method	model	techniqu	system	algorithm	inform	analysi	approach	structur
ASNetwork(BERT+all layers)	data	model	surfac	system	method	user	algorithm	volum	techniqu	visual

top-150 nodes in the *projection* view. The color is pre-defined by the k-means algorithm to determine the possible clusterings from the distances. To determine whether the close words are relevant, Selina brushes the nodes with the same colors to show the detailed local structures. The result is depicted in the Fig. 5, there are some dominant nodes with larger radius indicating higher *influence score* in the *Network* view. Based on her background knowledge, she correlates each cluster to the sub area as follows: **CL**(●): "lexi", "linguist", "sentenc" are related to linguistic language, and "sentiment" analysis is one of the common tasks in language analysis. **DS**(●): "graph", "sequenc", "text", "tree" refer to different types of data structures, and the word "algorithm" with the highest influence score also echoes the sub area. She confirms that the keywords from clustering(●) are more related to CV based on her domain knowledge. From these observations, she is positive that the word embeddings are good enough to capture domain-specific semantic meanings in the *projection* view.

Selina brushes the area around green words in the *projection* view and filters unimportant edges in the *network* view. She clicks the *Fix Graph* button to update the *community* view. As shown in Fig. 6(A), she identifies one community(●), the most important word is "imag" which shares many links with others. She further confirms these links are also meaningful, for example, *depth* and *texture* are two features of *image*, while *image segmentation* is a computer vision task to separate pixels into parts. She clicks the word "segment", and the *exploration* view can retrieve lots of segmentation related documents. Further, She filters the documents of interest by selecting the specific keyword relationship between "segment" and "semant". She receives some texts related to *semantic segmentation* which is one of the categories in CV segmentation task, shown in Fig. 6(B).

In another community(●), there are some dominant words which are also related to essential topics in CV, such as *recognition*, *speech*, *vision*. She confirms the link between "speech" and "vision" comes from some research ideas, which performs the speech analysis through computer vision by using Fourier transformation to convert speech to a spectrogram. "transform" is another keyword in this community, it has a high *influence score*, centrality score and low TF-IDF score. She infers that the reason might be its frequent appearance in many documents. The detailed properties can help her to infer the role of the keyword playing in the literature. In conclusion, Selina confirms that the results make sense based on her experience.

7 Discussion

7.1 Expert Feedback

We receive many valuable comments from 4 domain experts (**E 1-4**) in VIS. Also, we conduct an interview with an expert(**E5**) who has highly relevant expertise in Information Retrieval and Natural Language Processing.

Method: All the experts appreciated our novel idea to utilize self-attention to do keywords extraction. **E4**:"*I like the idea of using self-attention to extract themes in document corpora*". **E5** confirmed that it is important to do the fine-tuning process to learn domain-specific knowledge in NLP literature. Compared with traditional topic modeling, our KeywordMap can create dynamic and domain-specific keywords list having human in the loop. **E5** mentioned that our domain-specific keywords are useful and efficient in the Information Retrieval task. **E3** suggested us to evaluate the ASNetwork with the co-occurrence matrix to show the usefulness.

Visual Design: **E5** liked that our analytics flow and commented that our visual interface has good usability. **E2&E1** also confirmed our design tasks are well-defined and the visualization system is well-established. **E2** commented that "*I think this visualization system is useful to explore the set of keywords and meaningful documents correlated to the keywords.* "

7.2 Extensibility

Our work can potentially be extended to other problems. First, we evaluate how word clusters are formed when injecting domain-specific knowledge. This evaluation can be applied to other scenarios, for example, explaining the wrong predictions from the word clusters in explainable artificial intelligence. Second, identifying the influential keywords can be helpful to analyze other similar graphs from an information propagation point of view. Furthermore, our method proves the feasibility of making full use of existing labeled data to generate fine-grained topics. We hope our research can inspire new ideas in topic modeling.

7.3 Approach Limitations

KeywordMap also has several limitations. First, these pre-trained models require sub-word pieces tokenization to reduce the size of vocabulary, so we can not see some uncommon words in the final result. e.g. *convolutional* will be split into *"con","##vo","##vo","##al"*. Second, when the user performs graph pruning in the *network* view, KeywordMap lacks the capability to provide guidance for the proper parameter. Third, even assisted with zoom interaction, the *community* shows a limited number of communities due to space limitation.

8 Conclusion

In this paper, we propose to employ an attention mechanism to capture word relationships. Due to the lack of domain-specific knowledge in the pre-trained model, we apply a supervised classification task to incorporate domain-specific word relationships into the attention score. Inspired by a social network model, an attention-based word influence algorithm is presented to compute the word's importance. We evaluate the performance of our algorithm in a quantitative and qualitative manner. Furthermore, by carefully designing the requirements and tasks, we implement an interactive system, KeywordMap, for users to analyze keywords and perform information retrieval.

References

[1] F. Beck, S. Koch, and D. Weiskopf. Visual analysis and dissemination of scientific literature collections with survis. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):180–189, 2015.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[4] K. Börner, C. Chen, and K. W. Boyack. Visualizing knowledge domains. *Annual review of information science and technology*, 37(1):179–255, 2003.

[5] L.-F. Chien. Pat-tree-based keyword extraction for chinese information retrieval. In *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 50–58, 1997.

[6] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[7] T. Dang and V. T. Nguyen. Commodeler: Topic modeling using community detection. In *EuroVA@ EuroVis*, pp. 1–5, 2018.

[8] W. M. Darling, M. Paul, and F. Song. Unsupervised part-of-speech tagging in noisy and esoteric domains with a syntactic-semantic bayesian hmm. In *Proceedings of the Workshop on Semantic Analysis in Social Media*, pp. 1–9, 2012.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] T. L. Griffiths, M. Steyvers, D. M. Blei, and J. B. Tenenbaum. Integrating topics and syntax. In *Advances in neural information processing systems*, pp. 537–544, 2005.

[11] G. E. Hinton and S. T. Roweis. Stochastic neighbor embedding. In *Advances in neural information processing systems*, pp. 857–864, 2003.

[12] T. Hofmann. Probabilistic latent semantic analysis. *arXiv preprint arXiv:1301.6705*, 2013.

[13] P. Isenberg, F. Heimerl, S. Koch, T. Isenberg, P. Xu, C. D. Stolper, M. Sedlmair, J. Chen, T. Möller, and J. Stasko. vispubdata. org: A metadata collection about ieee visualization (vis) publications. *IEEE transactions on visualization and computer graphics*, 23(9):2199–2206, 2016.

[14] P. Isenberg, T. Isenberg, M. Sedlmair, J. Chen, and T. Möller. Toward a deeper understanding of visualization through keyword analysis. *arXiv preprint arXiv:1408.3297*, 2014.

[15] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[16] X. Jiang and J. Zhang. A text visualization method for cross-domain research topic mining. *Journal of Visualization*, 19(3):561–576, 2016.

[17] O. Kovaleva, A. Romanov, A. Rogers, and A. Rumshisky. Revealing the dark secrets of bert. *arXiv preprint arXiv:1908.08593*, 2019.

[18] T. K. Landauer, P. W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

[19] Z. Li, C. Zhang, S. Jia, and J. Zhang. Galex: Exploring the evolution and intersection of disciplines. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1182–1192, 2019.

[20] M. Litvak and M. Last. Graph-based keyword extraction for single-document summarization. In *Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization*, pp. 17–24. Association for Computational Linguistics, 2008.

[21] M. Liu, S. Liu, X. Zhu, Q. Liao, F. Wei, and S. Pan. An uncertainty-aware approach for exploratory microblog retrieval. *IEEE transactions on visualization and computer graphics*, 22(1):250–259, 2015.

[22] S. Liu, X. Wang, C. Collins, W. Dou, F. Ouyang, M. El-Assady, L. Jiang, and D. A. Keim. Bridging text visualization and mining: A task-driven survey. *IEEE transactions on visualization and computer graphics*, 25(7):2482–2504, 2018.

[23] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[24] K. K. Mane and K. Börner. Mapping topics and topic bursts in pnas. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5287–5290, 2004.

[25] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[26] O. Medelyan, E. Frank, and I. H. Witten. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pp. 1318–1327, 2009.

[27] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *Proceedings of the 17th international conference on World Wide Web*, pp. 101–110, 2008.

[28] R. Mihalcea and P. Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404–411, 2004.

[29] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[30] R. Misra. News category dataset, 06 2018. doi: 10.13140/RG.2.2.20331.18729

[31] T. Morita. Keyword associative document retrieval system, Mar. 22 1994. US Patent 5,297,042.

[32] N. S. F. (NSF). *Science and engineering indicators 2010*. ERIC Clearinghouse, 2010.

[33] I. U. Ogul, C. Ozcan, and O. Hakdagli. Keyword extraction based on word synonyms using word2vec. In *2019 27th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4. IEEE, 2019.

[34] A. Onan, S. Korukoğlu, and H. Bulut. Ensemble of keyword extraction methods and classifiers in text classification. *Expert Systems with Applications*, 57:232–247, 2016.

[35] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[36] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[37] D. M. Romero, W. Galuba, S. Asur, and B. A. Huberman. Influence and passivity in social media. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 18–33. Springer, 2011.

[38] M. Rosvall and C. T. Bergstrom. Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. *PloS one*, 6(4):e18209, 2011.

[39] M. Sanderson. Christopher d. manning, prabhakar raghavan, hinrich schütze, introduction to information retrieval, cambridge university press. 2008. isbn-13 978-0-521-86571-5, xxi+ 482 pages. *Natural Language Engineering*, 16(1):100–103, 2010.

[40] M. Selvi, K. Thangaramya, M. Saranya, K. Kulothungan, S. Ganapathy, and A. Kannan. Classification of medical dataset along with topic modeling using lda. In *Nanoelectronics, Circuits and Communication Systems*, pp. 1–11. Springer, 2019.

[41] Q. Shen, T. Wu, H. Yang, Y. Wu, H. Qu, and W. Cui. Nameclarifier: A visual analytics system for author name disambiguation. *IEEE transactions on visualization and computer graphics*, 23(1):141–150, 2016.

[42] D. Suleiman, A. A. Awajan, and W. Al Etaiwi. Arabic text keywords extraction using word2vec. In *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, pp. 1–7. IEEE, 2019.

[43] N. J. Van Eck and L. Waltman. Text mining and visualization using vosviewer. *arXiv preprint arXiv:1109.2058*, 2011.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[45] E. M. Voorhees and D. Harman. Overview of the sixth text retrieval conference (trec-6). *Information Processing & Management*, 36(1):3–35, 2000.

[46] X. Wang, S. Liu, Y. Chen, T.-Q. Peng, J. Su, J. Yang, and B. Guo. How ideas flow across multiple social groups. In *2016 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pp. 51–60. IEEE, 2016.

[47] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. Kea: Practical automated keyphrase extraction. In *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*, pp. 129–152. IGI Global, 2005.

[48] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pp. 5753–5763, 2019.

[49] C.-K. Yau, A. Porter, N. Newman, and A. Suominen. Clustering scientific documents with topic modeling. *Scientometrics*, 100(3):767–786, 2014.