National University of Computer and Emerging Sciences, Karachi

Literature & Application Review

Automation of Minutes of Meeting

Author:
Mehdi Raza K16-3904,
Hussain Yousuf K16-3805,
Shehryar Naeem K16-3950

Supervisor:
Dr. Muhammad Rafi

22nd December, 2019

Research Paper #1:

Title:

Improved Keyword and Keyphrase Extraction from Meeting Transcripts

Reference:

Sheeba, J. I., and K. Vivekanandan. "Improved keyword and keyphrase extraction from meeting transcripts." International Journal of Computer Applications 52.13 (2012).

Summary:

The aim of this paper is to extract low-frequency keywords and keyphrases for every sentence in the meeting transcripts. In this unsupervised approach, Maximum Entropy (MaxEnt) and SVM classifiers are used to determine whether a word is a keyword, bigram and trigram keywords are retrieved using N-gram based approach. In addition, low-frequency keywords are identified using LDA (Latent Dirichlet allocation) model. The MaxEnt classifier functions on the following formula:

$P(x) = \exp(c_1 * f_1(x) + c_2 * f_2(x) + c_3 * f_3(x) \ldots) / Z$

Sequential pattern mining is further used to improve the quality of extracted keywords also double-quoted and capitalized words are too included as keywords. The experimental result suggests that the MaxEnt classifier shows better performance when compared with the SVM classifier. However, through the LDA approach, the number of keywords extracted was increased.

Research Paper #2:

Title:

A Supervised Framework for Keyword Extraction From Meeting Transcripts

Reference:

Liu, Fei, Feifan Liu, and Yang Liu. "A supervised framework for keyword extraction from meeting transcripts." IEEE Transactions on Audio, Speech, and Language Processing 19.3 (2010): 538-548.

Summary:

This paper presents a supervised framework for extracting keywords from meeting transcripts. The paper claims to achieve better performance than TF-IDF. ICSI meeting corpus is used as training data which consists of naturally occurring meeting recordings, each about an hour long. All the meetings have been transcribed and annotated with dialogue acts, topic boundaries, and extractive summaries. For the supervised framework, a maximum
entropy classifier is used. To measure the effectiveness of different features, three feature selection processes are performed namely;
Forward feature selection: It begins with an empty set, and iteratively adds the feature that achieves the largest performance gain when combined with the currently selected features.
Backward feature selection. It starts with the entire feature set and removes one feature in each iteration to have the least performance degradation.
Dynamic programming (DP) based feature selection: Similar to forwarding feature selection, this approach also starts with an empty set. Unlike forward and backward methods that only keep the best feature subset in each iteration, this approach maintains best feature subsets from
a set of features in each iteration using a dynamic programming-based strategy.
Experimentation shows that the supervised framework with confidence score-based summaries achieves the best performance, and our proposed method outperforms the unsupervised TF-IDF weighting and a state-of-the-art keyphrase extraction system (Kea) on all testing conditions.

Research Paper #3:

Title:

Rapid Keyphrase Extraction from Audio Transcripts of Video Lectures

Reference:

Balagopalan, Arun, et al. "Automatic keyphrase extraction and segmentation of video lectures." 2012 IEEE International Conference on Technology Enhanced Education (ICTEE). IEEE, 2012.

Summary:

In this paper, we use a supervised machine learning algorithm to train the domain-specific features. In addition, the decision tree classifier model is used to enhance the efficiency of the feature engineering and to extract relevant key phrases. Preprocessing steps includes
POS TAGGING:
The noun tags(NN, NNS, NP), verb tags(VBZ, VBG, VBN) and adjective tags(JJ) are the most important tags. UPenn TreeBank II tagset is used for POS tagging.
STOP WORDS REMOVAL:
The common words such as prepositions, articles, apostrophes, brackets, numbers and non-alphanumeric that are irrelevant to the context are removed. Hyphens are kept, hyphenated words are considered to be single words.
N-GRAM EXTRACTION:
The input text is split up according to phrase boundaries. N-Gram takes a list of words and builds a set of consecutive word pairs.
STEMMING:
The goal of stemming or lemmatization is to reduce a word to its common base form. Thus the key terms of a lecture are represented by stems instead of original words.

The system uses a supervised machine learning algorithm to train and test datasets using a decision tree classifier model. Gini index is used along with the ID3 algorithm for the classification and creation of a decision tree. The dataset is trained using the ID3 algorithm to produce a decision tree for identifying keyphrases. Gini index is used to calculate the possible split values of attributes.
Experimentation shows that the Decision tree model overcomes the drawback of Naive Bayes classifier that assumes feature independence.

Research Paper #4:

Title:

The Probability Ranking Principle in IR

Reference:

Robertson, Stephen E. "The probability ranking principle in IR." Readings in information retrieval (1997): 281-286.

Summary:

The paper described one of the most used Ranking Principle i.e. the Probability Ranking Principle which is mainly based on axioms of probability and precisely on **Bayes Rule** i.e. the probability of an event, based on prior knowledge of conditions that might be related to the event.

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

A, B = events
P(A|B) = probability of A given B is true
P(B|A) = probability of B given A is true
P(A) P(B) = the independent probabilities of A and B

The paper stated that;
"If a reference retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing probability of relevance to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data have been made available to the system for this purpose, the overall effectiveness of the system to its user will be the best that is obtainable on the basis of those data."
Further, the paper introduces the following formula;
Let x be a document in the collection.
Let R represent the relevance of a document w.r.t. given (fixed) query and let NR represent non-relevance.
Need to find p(R|x) - the probability that retrieved document x is relevant.

$$p(R \mid x) = \frac{p(x \mid R)p(R)}{p(x)}$$

$$p(NR \mid x) = \frac{p(x \mid NR)p(NR)}{p(x)}$$

p(x|R), p(x|NR) - probability that if a relevant (non-relevant) The document is retrieved, it is x.
p(R),p(NR) - prior probability of retrieving a (non) relevant document

$$p(R \mid x) = \frac{p(x \mid R)\,p(R)}{p(x)}$$

$$p(NR \mid x) = \frac{p(x \mid NR)\,p(NR)}{p(x)}$$

Ranking Principle (Bayes' Decision Rule):
If p(R|x) > p(NR|x) then x is relevant, otherwise, x is not relevant

After the derivation the final formula derived for Retrieval Status Value (RSV) was;

$$RSV_d = \log \prod_{t:x_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)} = \sum_{t:x_t=q_t=1} \log \frac{p_t(1-u_t)}{u_t(1-p_t)}$$

where;

$$p_t = \frac{|V_t| + \frac{1}{2}}{|V| + 1} \qquad u_t = \frac{\mathrm{df}_t - |V_t| + \frac{1}{2}}{N - |V| + 1}.$$

Research Paper #5:

Title:

Text categorization with Support Vector Machines: Learning with many relevant features

Reference:

Joachims, Thorsten. "Text categorization with support vector machines: Learning with many relevant features." European conference on machine learning. Springer, Berlin, Heidelberg, 1998.

Summary:

The paper discusses the working and benefits of commonly used text categorization technique i.e. Support Vector Machine. Firstly, the paper describes the steps of preparing data for categorization that mainly includes, tokenization, stemming, stop word removal, computing term, and inverse document frequency and selecting the words with minimum three occurrences in the corpus.

Then the paper defines the working of SVM which i.e. in this algorithm we plot each data item as a point in n-dimensional space (where n is the number of features - in this case, features are distinct words) with the value of each feature being the value of a particular coordinate. Then, the algorithm performs classification by determining the hyper-plane that distinguishes the two groups very well. The SVM finds the hypothesis which minimizes the bound on the true error.

The paper also defends that SVM works well for text classification because its performance is based on the margin that separates the data nor on a number of features as the text has a huge number of features. Also, there are few irrelevant features so we can easily drop them. Moreover, as the document vectors are too sparse which supports SVM. Most importantly the most text categorization problems are linearly separable so SVM best fits in this case.

Research Paper #6:

Title:

Speaker Diarization With LSTM

Reference:

Wang, Quan, et al. "Speaker diarization with LSTM." 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018.

Summary:

This paper uses LSTM-based d-vector audio embeddings with spectral clustering to solve the speaker diarization problem. This method comprises of 3 steps:

1. Speech segmentation: where the input audio is segmented into short sections that are assumed to have a single speaker, and the non-speech sections are filtered out.
2. Audio embedding extraction: where specific features such as MFCCs, d-vectors are extracted.
3. Clustering: where the number of speakers is determined, and the extracted audio embeddings are clustered into these speakers.

The d-vector model is a 3-layer LSTM network with a final linear layer. Each LSTM layer has 768 nodes, with a projection of 256 nodes.
The spectral clustering algorithm consists of the following steps:

1. Construct the affinity matrix A, where Aij is the cosine similarity between the ith and the jth segment embedding.
2. Apply the following sequence of refinement operations on the affinity matrix A:
   Gaussian Blur   Row-wise Thresholding   Symmetrization
   Diffusion   Row-wise Max Normalization
   These refinements act to both smooth and denoise the data and are crucial to the success of the algorithm.
3. Perform Eigen-Decomposition on the refined affinity matrix.
4. Then we use the K-Means algorithm to cluster these new embeddings.

Two clustering types are discussed:
Online clustering: A speaker label is immediately emitted once a segment is available.
Particularly we apply a threshold on the similarities between embeddings of segments. cosine similarity is used as our similarity metric.
Offline clustering: Speaker labels are produced after the embeddings of all segments are available.

Review of Application #1:

Name of Application:

Reason8

URL Link:

https://reason8.ai/

Summary:

- General Overview:
    - Each participant in the room needs the app on their phone.
    - IOS and Android Mobile Application no desktop version.
- Working:
    - One participant presses the ‘Host meeting’ in the app.
    - A meeting ID number is displayed.
    - Other participants press ‘Join’ and type this meeting ID number when prompted.
    - The phones stay on the desk near their owners.
    - Someone presses ‘Start’ to kick the recording off when you are ready to begin.
    - The recording happens in the background.
    - When the meeting is over, stop the recording.
    - The recording is accessible via the website.
- Evaluation Criteria:
    - Accuracy of transcription: Good.
    - Accuracy of speaker attribution: Good.
    - Output: Free text editor means you can create ‘real’ meeting notes directly in the tool.
- Other Features:
    - Search within a transcription
    - Highlight key phrases during recording
    - Prepare summaries
    - Share with colleagues
    - Download your recording
    - Download text
    - Low Learning curve and High Time-saving rating

Review of Application #2:

Name of Application:

Otter

URL Link:

https://otter.ai/

Summary:

- General Overview:
    - Otter is a note-taking app that also records in-person meetings.
    - You only need to have it on one device, so it doesn't feel like it should get good results when you've got a lot of people in the room.
    - It identifies speakers, lets you highlight key phrases and creates a transcript in real-time that you can see on the screen.
    - You can take photos during your recording (on the app version). I could snap a photo of the notes on the whiteboard to embed in the meeting notes.
    - You can import audio or video to be transcribed.
- Evaluation Criteria:
    - Accuracy of transcription: Good.
    - Accuracy of speaker attribution: Poor, partly because having only one device meant the participant who was furthest away struggled to be heard.
    - Output: Transcript needs to be edited in another tool to create 'real' minutes.
- Other Features:
    - Records in the browser
    - Google and Zoom Integration
    - Search within a transcription
    - Share with colleagues
    - Download your recording
    - Download text
    - Low learning curve
    - Image capture
    - High Time Saving Rate

Review of Application #3:

Name of Application:

Voicea

URL Link:

https://www.voicea.com/

Summary:

- General Overview:
  - Uses an AI assistant as a way to capture your meeting notes.
  - Add your agenda so that the software knows what words to look for. I think you'd need to do this before the meeting to get the benefit.
  - You can 'favorite' a recording if you want it to stand out in the overall list of meetings.
  - The Insights tab pulls out things the software thinks might be important, like mentions of sales, points of contention, locations mentioned and so on. This was largely irrelevant to the meeting I held, but it might add some value to other types of discussion.
- Evaluation Criteria:
  - Accuracy of transcription: Poor, in comparison to others.
  - Accuracy of speaker attribution: Good. Voicea splits the transcript up by phrase, so you can tell different phrases were said by different people but don't attribute their names.
  - Output: Lots of support from the tool with highlights and insights, but you would need to practice using the highlight features to get the best transcript. We would need to create 'real' minutes in another tool.
- Other Features:
  - Google, Office 365, Salesforce, Slack, BlueJeans Integration
  - Harder to setup
  - Search within a transcription
  - Highlight key phrases during recording
  - Share with colleagues
  - Download text
  - Medium Learning curve

Review of Application #4:

Name of Application:

TERMINUTER

Summary:

- General Overview:
  - Automated cognitive meeting minutes tool, mainly based on speech-to-text technology
  - The app automatically writes and structures meeting minutes with decisions and to-dos and even alerts you if owners or deadlines are not defined
  - Offers a search function for past meeting minutes
  - Uses Watson Speech-To-Text as well as other cognitive applications to turn your meeting notes into structured minutes