16 Text Summarization

Text summarization refers to the task of compressing a relatively large amount of text data or a long text article into a more concise form for easy digestion. It is obviously very important for text data access, where it can help users see the main content or points in the text data without having to read all the text. Summarization of search engine results is a good example of such an application. However, summarization can also be useful for text data analysis as it can help reduce the amount of text to be processed, thus improving the efficiency of any analysis algorithm.

However, summarization is a non-trivial task. Given a large document, how can we convey the important points in only a few sentences? And what do we mean by "document" and "important"? Although it is easy for a human to recognize a good summary, it is not as straightforward to define the process. In short, for any text summarization application, we'd like a semantic compression of text; that is, we would like to convey essentially the same amount of information in less space. The output should be fluent, readable language. In general, we need a purpose for summarization although it is often hard to define one. Once we know a purpose, we can start to formulate how to approach the task, and the problem itself becomes a little easier to evaluate.

In one concrete example, consider a news summary. If our input is a collection of news articles from one day, a potentially valid output is a list of headlines. Of course, this wouldn't be the entire list of headlines, but only those headlines that would interest a user. For a different angle, consider a news summarization task where the input is one text news article and the output should be one paragraph explaining what the article talks about in a readable format. Each task will require a different solution.

Summarizing retrieval results is also of particular interest. On a search engine result page, how can we help the user click on a relevant link? A common strategy is to highlight words matching the query in a short snippet. An alternative approach would be to take a few sentences to summarize each result and display the short

summaries on the results page. Using summaries in this way could give the user a better idea of what information the document contains before he or she decides to read it.

Opinion summarization is useful for both businesses and shoppers. Summarizing all reviews of a product lets the business know whether the buyers are satisfied (and why). The review summaries also let the shoppers make comparisons between different products when searching online. Reviews can be further broken down into summaries of positive reviews and summaries of negative reviews. An even more granular approach described in Wang et al. [2010] and Wang et al. [2011] and further discussed in Chapter 18 uses topic models to summarize product reviews relating to different aspects. For hotel reviews, this could correspond to service, location, price, and value. Although the output in these two works is not a human-readable summary, we could imagine a system that is able to summarize all the hotel reviews in English (or any other language) for the user.

In this chapter, we overview two main paradigms of summarization techniques and investigate their different applications.

16.1 Overview of Text Summarization Techniques

There are two main methods in text summarization. The first is **selection-based** or **extractive summarization**. With this method, a summary consists of a sequence of sentences selected from the original documents. No new sentences are written, hence the summary is *extracted*. The second method is **generation-based** or **abstractive summarization**. Here, a summary may contain new sentences not in any of the original documents. One method that we explore here is using a language model. Previously in this book, we've used language models to calculate the likelihood of some text; in this chapter, we will show how to use a language model in reverse to generate sentences. We also briefly touch on the field of **natural language generation** in our discussion of abstractive techniques.

Following the pattern of previous chapters, we then move on to evaluation of text summarization. The two methods each have evaluation metrics that are particularly focused towards their respective implementation, but it is possible to use (e.g.) an abstractive evaluation metric on a summary generated by an extractive algorithm. Finally, we look into some applications of text summarization and see how they are implemented in real-world systems.

Text summarization is a broad field and we only touch on the core concepts in this chapter. For further reading, we recommend that the reader start with Das and Martins [2007], which provides a systematic overview of the field and contains much of the content from this chapter in an expanded form.

16.2 Extractive Text Summarization

Information retrieval-based techniques use the notion of sentence vectors and similarity functions in order to create a summarization text. A sentence vector is equivalent in structure to a document vector, albeit based on a smaller number of words. Below, we will outline a basic information retrieval-based summarization system.

1. Split the document to be summarized into sections or passages.
2. For each passage, "compress" its sentences into a smaller number of relevant (yet not redundant) sentences.

This strategy retains coherency since the sentences in the summary are mostly in the same order as they were in the original document.

Step one is portrayed in Figure 16.1. The sentences in the document are traversed in order and a normalized, symmetric similarity measure (see Chapter 14) is applied on adjacent pairs of sentences. The plot on the right-hand side of the figure shows the change in similarity between the sentences. We can inspect these changes to segment the document into passages when the similarity is low, i.e., a shift in topic occurs. An alternative approach to this segmentation is to simply use paragraphs if the document being operated on contains that information, although most of the time this is not the case. This rudimentary partitioning strategy is a task in

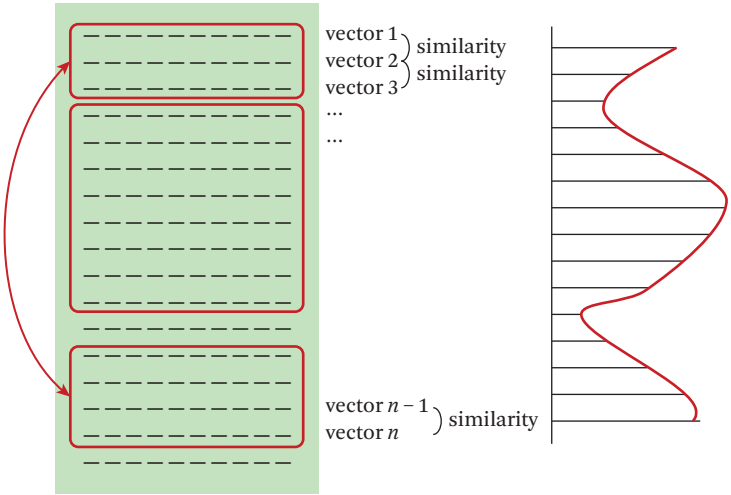


Figure 16.1 Segmenting a document into passages with a similarity-based discourse analysis.

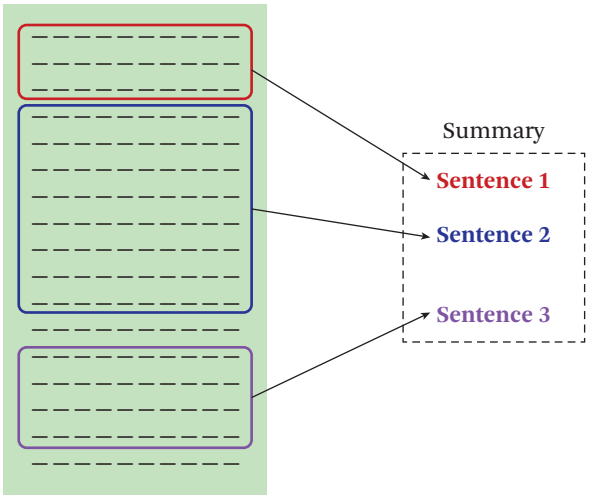


Figure 16.2 **Text summarization using maximum marginal relevance to select one sentence from each passage as a summary.**

discourse analysis (a subfield of NLP). Discourse analysis deals with sequences of sentences as opposed to only one sentence.

Now that we have our passages, how can we remove redundancy and increase diversity in the resulting summarization during step two? The technique **maximal marginal relevance** (MMR) reranking can be applied to our problem. Essentially, this algorithm greedily reranks each sentence in the current passage, outputting only the top few as a summary. Figure 16.2 shows the output of the algorithm when we only select one sentence from each passage.

The MMR algorithm is as follows. Assume we are given an original list R and a profile p to construct the set of selected sentences S (where $|S| \ll |R|$). R is a partitioned chunk of sentences in the document we wish to summarize. The profile p determines what is exactly meant by "relevance." Originally, the MMR formula was applied to documents returned from an information retrieval system (hence the term *reranking*). Documents were selected based on their marginal relevance to a query (which is our variable p) in addition to non-redundancy to already-selected documents. Since our task deals with sentence retrieval, p can be a user profile (text about the user), the entire document itself, or it could even be a query formulated by the user.

According to marginal relevance, the next sentence s_i to be added into the selected list S is defined as

$$s_i = \arg\max_{s \in R \setminus S} \left\{ (1-\lambda) \cdot \text{sim}_1(s, p) - \lambda \cdot \arg\max_{s_j \in S} \text{sim}_2(s, s_j) \right\}. \quad (16.1)$$

The $R \setminus S$ notation may be read as "R set minus S", i.e., all the elements in R that are not in S. The MMR formulation uses $\lambda \in [0, 1]$ to control relevance versus redundancy; the positive relevance score is discounted by the amount of redundancy (similarity) to the already-selected sentences. Again, the two similarity metrics may be any normalized, symmetric measures. The simplest instantiation for the similarity metric would be cosine similarity, and this is in fact the measure used in Carbonell and Goldstein [1998].

The algorithm may be terminated once an appropriate number of words or sentences is in S, or if the score $\text{sim}_1(s, p)$ is below some threshold. Furthermore, the similarity functions may be tweaked as well. Could you think of a way to include sentence position in the similarity function? That is, if a sentence is far away (dissimilar) from the candidate sentence, we could subtract from the similarity score. Even better, we could interpolate the two values into a new similarity score such as

$$\text{sim}(s, s') = \alpha \cdot \text{sim}_{\text{cosine}}(s, s') + (1-\alpha) \cdot \left(1 - \frac{d(s, s')}{\max d(s, \cdot)}\right), \quad (16.2)$$

where $\alpha \in [0, 1]$ controls the weight between the regular cosine similarity and the distance measure, and $d(\cdot, \cdot)$ is the number of sentences between the two parameters. Note the "one minus" in front of the distance calculation, since a smaller distance implies a greater similarity.

Of course, λ in the MMR formula is also able to be set. In fact, for multi-document summarization, Das and Martins [2007] suggests starting out with $\lambda = 0.3$ and then slowly increasing to $\lambda = 0.7$. The reasoning behind this is to first emphasize novelty and then default to relevance. This should remind you of the exploration-exploitation tradeoff discussed in Chapter 11.

16.3 Abstractive Text Summarization

An abstractive summary creates sentences that did not exist in the original document or documents. Instead of a document vector, we will use a language model to represent the original text. Unlike the document vector, our language model gives us a principled way in which to generate text. Imagine we tokenized our document with unigram words. In our language model, we would have a parameter representing the probability of each word occurring. To create our own text, we will draw words from this probability distribution.

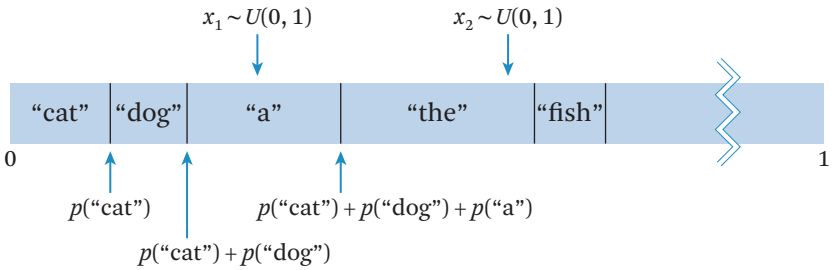


Figure 16.3 **Drawing words from a unigram language model.**

Say we have the unigram language model θ estimated on a document we wish to summarize. We wish to draw words $w_1, w_2, w_3, \ldots$ from θ that will comprise our summary. We want the word w_i to occur in our summary with about the same probability it occurred in the original document—this is how our generated text will approximate the longer document. Figure 16.3 depicts how we can accomplish this task. First, we create a list of all our parameters and incrementally sum their probabilities; this will allow us to use a random number on $[0, 1]$ to choose a word w_i. Simply, we get a uniform random floating point number between zero and one. Then, we iterate through the words in our vocabulary, summing their probabilities until we get to the random number. We output the term and repeat the process.

In the example, imagine we have the following values:

$p(\text{cat})$	0.010
$p(\text{cat}) + p(\text{dog})$	0.018
$p(\text{cat}) + p(\text{dog}) + p(\text{a})$	0.038
$\vdots$	$\vdots$
$p(\text{cat}) + p(\text{dog}) + p(\text{a}) + \cdots + p(\text{zap})$	1.0

Say we generate a random number x_1 using a uniform distribution on $[0, 1]$. This is denoted as $x_1 \sim \mathcal{U}(0, 1)$. Now imagine that $x_1 = 0.032$. We go to the cumulative point 0.032 in our distribution and output "a". We can repeat this process until our summary is of a certain length or until we generate an end-of-sentence token `</s>`.

At this point, you may be thinking that the text we generate will not make any sense—that is certainly true if we use a unigram language model since each word is generated independently without regard to its context. If more fluent language is required, we can use an n-gram language model, where $n > 1$. Instead of each word being independently generated, the new word will depend on the previous $n - 1$

words. The generation will work the same way as in the unigram case: say we have the word w_i and wish to generate w_{i+1} with a bigram language model. Our bigram language model gives us a distribution of words that occur after w_i and we draw the next word from there in the same way depicted in Figure 16.3.

The sentence generation from a bigram language model proceeds as follows: start with (e.g.) *The*. Then, pick from the distribution $p(w \mid \text{The})$ using the cumulative sum technique. The next selected word could be *cat*. Then, we use the distribution $p(w \mid \text{cat})$ to find the next w, and so on. While the unigram model only had one "sum table" (Figure 16.3), the bigram case needs V tables, one for each w' in $p(w \mid w')$.

Typically, the n-value will be around three to five depending on how much original data there is. We saw what happened when n is too small; we get a jumble of words that don't make sense together. But we have another problem if n is too large. Consider the extreme case where $n = 20$. Then, given 19 words, we wish to generate the next one using our 20-gram language model. It's very unlikely that those 19 words occurred more than once in our original document. That means there would only be one choice for the 20th word. Because of this, we would just be reproducing the original document, which is not a very good summary. In practice, we would like to choose an n-gram value that is large enough to produce coherent text yet small enough to not simply reproduce the corpus.

There is one major disadvantage to this abstractive summarization method. Due to its nature, a given word only depends on the n surrounding words. That is, there will be no long-range dependencies in our generated text. For example, consider the following sentence generated from a trigram language model:

They imposed a gradual smoking ban on virtually all corn seeds planted are hybrids.

All groups of three words make sense, but as a whole the sentence is incomprehensible; it seems the writer changed the topic from a smoking ban to hybrid crops mid-sentence. In special cases, when we restrict the length of a summary to a few words when summarizing highly redundant text, such a strategy appears to be effective as shown in the micropinion summarization method described in Ganesan et al. [2012].

16.3.1 Advanced Abstractive Methods

Some advanced abstractive methods rely more heavily on natural language processing to build a model of the document to summarize. **Named entity recognition** can be used to extract people, places, or businesses from the text. **Dependency parsers** and other syntactic techniques can be used to find the relation between the entities

and the actions they perform. Once these actors and roles are discovered, they are stored in some internal representation. To generate the actual text, some representations are chosen from the parsed collection, and English sentences are created based on them; this is called **realization**.

Such realization systems have much more fine-grained control over the generated text than the basic abstractive language model generator described above. A templated document structure may exist (such as intro→paragraph 1→paragraph 2→conclusion), and the structures are chosen to fill each spot. This control over text summarization and layout enables an easily-readable summary since it has a natural topical flow. In this environment, it would be possible to merge similar sentences with conjunctions such as *and* or *but*, depending on the context. To make the summary sound even more natural, pronouns can be used instead of entity names if the entity name has already been mentioned. Below are examples of these two operations:

> *Gold prices fell today. Silver prices fell today.* → *Gold and silver prices fell today.*
> *Company A lost* 9.43% *today. Company A was the biggest mover.* → *Company A lost* 9.43% *today. It was the biggest mover.*

Even better would be

> *Company A was today's biggest mover, losing* 9.43%.

These operations are possible since the entities are stored in a structured format. For more on advanced natural language generation, we suggest Reiter and Dale [2000], which has a focus on practicality and implementation.

16.4 Evaluation of Text Summarization

In extractive summarization, representative sentences were selected from passages in the text and output as a summary. This solution is modeled as an information retrieval problem, and we can evaluate it as such. Redundancy is a critical issue, and the MMR technique we discussed attempts to alleviate it. When doing our evaluation, we should consider redundant sentences to be irrelevant, since the user does not want to read the same information twice. For a more detailed explanation of IR evaluation measures, please consult Chapter 9.

For full output scoring, we should prefer IR evaluation metrics that do not take into account result position. Although our summary is generated by ranked sentences per passage, the entire output is not a ranked list since the original document is composed of multiple passages. Therefore we can use precision, recall, and F_1 score.

It is possible to rank the passage scoring retrieval function using position-dependent metrics such as average precision or NDCG, but with the final output this is not feasible. Thus we need to decide whether to evaluate the passage scoring or the entire output (or both). Entire output scoring is likely more useful for actual users, while passage scoring could be useful for researchers to fine-tune their methods.

In abstractive summarization, we can't use the IR measures since we don't have a fixed set of candidate sentences. How can we compute recall if we don't know the total number of relevant sentences? There is also no intermediate ranking stage, so we also can't use average precision or NDCG (and again, we don't even know the complete set of correct sentences).

A laborious yet accurate evaluation would have human annotators create a gold standard summary. This "perfect" summary would be compared with the generated one, and some measure (e.g., ROUGE) would be used to quantify the difference. For the comparison measure, we have many possibilities—any measure that can compare two groups of text would be potentially applicable. For example, we can use the cosine similarity between the gold standard and generated summary. Of course, this has the downside that fluency is completed ignored (using unigram words). An alternative means would be to learn an n-gram language model over the gold standard summary, and then calculate the log-likelihood of the generated summary. This can ensure a basic level of fluency at the n-gram level, while also producing an interpretable result. Other comparisons between two probability distributions would also be applicable, such as KL-divergence.

The overall effectiveness of a summary can be tested if users read a summary and then answer questions about the original text. Was the summary able to capture the important information that the evaluator needs? If the original text was an entire textbook chapter, could the user read a three-paragraph summary and obtain sufficient information to answer the provided exercises? This is the only metric that can be used for both extractive and abstractive measures. Using a language model to score an extractive summary vs. an abstractive one would likely be biased towards the extractive one since this method contains phrases directly from the original text, giving it a very high likelihood.

16.5 Applications of Text Summarization

At the beginning of the chapter, we've already touched on a few summarization applications; we mentioned news articles, retrieval results, and opinion summarization. Summarization saves users time from manually reading the entire corpus while simultaneously enhancing preexisting data with summary "annotations."

The aspect opinion analysis mentioned earlier segments portions of user reviews into speaking about a particular topic. We can use this topic analysis to collect passages of text into a large group of comments on one aspect. Instead of describing this aspect with sorted unigram words, we could run a summarizer on each topic, generating readable text as output. These two methods complement each other, since the first step finds what aspects the users are interested in, while the second step conveys the information.

A theme in this book is the union of both structured and unstructured data, mentioned much more in detail in Chapter 19. Summarization is an excellent example of this application. For example, consider a financial summarizer with text reports from the Securities and Exchange Commission (SEC) as well as raw stock market data. Summarizing both these data sources in one location would be very valuable for (e.g.) mutual fund managers or other financial workers. Being able to summarize (in text) a huge amount of structured trading data could reveal patterns that humans would otherwise be unaware of—this is an example of **knowledge discovery**.

E-discovery (electronic discovery) is the process of finding relevant information in litigation (lawsuits and court cases). Lawyers rely on e-discovery to sift through vast amounts of textual information to build their case. The Enron email dataset[1] is a well-known corpus in this field. Summarizing email correspondence between two people or a department lets investigators quickly decide whether they'd like to dig deeper in a particular area or try another approach. In this way, summarization and search are coupled; search allows a subset of data to be selected that is relevant to a query, and the summarization can take the search results and quickly explain them to the user. Finally, linking email correspondence together (from sender to receivers) is a structured complement to the unstructured text content of the email itself.

Perhaps of more interest to those reading this book is the ability to summarize research from a given field. Given proceedings from a conference, could we have a summarizer explain the main trends and common approaches? What was most novel compared to previous conferences? When writing your own paper, can you write everything except the introduction and related work? The introduction is an overview summary of your paper. Related work is mostly a summary of papers similar to yours.

1. https://www.cs.cmu.edu/~./enron/

Bibliographic Notes and Further Reading

As mentioned in this chapter, Das and Martins [2007] is a comprehensive survey on summarization techniques. Additionally, Nenkova and McKeown [2012] is a valuable read. For applications, latent aspect rating analysis [Wang et al. 2010], [Wang et al. 2011] is a form of summarization applied to product reviews. We mention this particular application in more detail in Chapter 18. A typical extractive summarizer is presented in Radev et al. [2004], a typical abstractive summarizer is presented in Ganesan et al. [2010], and evaluation suggestions are presented in Steinberger and Jezek [2009]. The MMR algorithm was originally described in Carbonell and Goldstein [1998]. For advanced NLG (natural language generation) techniques, a good starting point is Reiter and Dale [2000].

Exercises

16.1. Do you think one summarization method (extractive or abstractive) would perform better on a small dataset? How about a large dataset? Justify your reasoning.

16.2. Explain how you can improve the passage detection by looking beyond only the adjacent sentences. How would you implement this?

16.3. Write a basic passage segmenter in MeTA. As input, take a document and extract the sentences into a vector with a built-in tokenizer. Segment the vector into passages using a similarity algorithm.

16.4. Now that you have a document segmented into passages, use MeTA to set up a search engine over each passage, where you treat passages as individual documents. Ensure that you have enough sentences per passage. You many need to tweak your previous answer to achieve this.

16.5. With your passage search engine, find a representative sentence from each passage to create a summary for the original document.

16.6. Use MeTA's language model to learn a distribution of words over a document you wish to summarize.

16.7. Add a `generate` function to the language model. It should take a context ($n-1$ terms) and generate the n^{th} term. Use the calculation described in this chapter to generate the next word.

16.8. Summarize the input document using the generator. Experiment with different stopping criteria. Which seems to work the best?

16.9. Create some simple post-processing rules for natural language generation realization. The examples we gave in the text were sentence joining and pronoun insertion. What else can you think of?

16.10. Explain how we can combine text summarization and topic modeling to create a powerful exploratory text mining application.

16.11. What can we accomplish by interpolating a language model distribution for an abstractive summarizer with another probability distribution, perhaps from existing summaries?

References

C. C. Aggarwal. 2015. *Data Mining - The Textbook*. Springer. DOI: 10.1007/978-3-319-14142-8. 296

C. C. Aggarwal and C. Zhai, editors. 2012. *Mining Text Data*. Springer. DOI: 10.1007/978-1-4614-3223-4. 296, 315

J. Allen. 1995. *Natural Language Understanding*. 2nd ed. Benjamin-Cummings Publishing Co., Inc., Redwood City, CA. 54

G. Amati and C. J. Van Rijsbergen. October 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389. DOI: 10.1145/582415.582416. 87, 88, 90, 111

A. U. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. 2009. On smoothing and inference for topic models. In *UAI 2009, Proc. of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada*, June 18-21, 2009, pp. 27–34. 385

R. A. Baeza-Yates and B. A. Ribeiro-Neto. 2011. *Modern Information Retrieval - the concepts and technology behind search*. 2nd ed. Pearson Education Ltd., Harlow, UK. http://www.mir2ed.org/. xvii, 18, 19

Y. Bar-Hillel, *The Present Status of Automatic Translation of Languages*, in Advances in Computers, vol. 1 (1960), pp. 91–163.

R. Belew. 2008. *Finding Out About: A Cognitive Perspective on Search Engine Technology and the WWW*. Cambridge University Press. 18

N. J. Belkin and W. B. Croft. 1992. Information filtering and information retrieval: Two sides of the same coin? *Commun. ACM*, 35(12):29–38. DOI: 10.1145/138859.138861. 84

C. M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ. 19, 37, 312, 385, 462

D. M. Blei, A. Y. Ng, and M. I. Jordan. March 2003. Latent Dirichlet Allocation. *J. of Mach. Learn. Res.*, 3:993–1022. 385

J. S. Breese, D. Heckerman, and C. Kadie. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, Morgan Kaufmann Publishers Inc. pp. 43–52, San Francisco, CA. http://dl.acm.org/citation.cfm?id=2074094.2074100. 235

P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. Della Pietra, and J. C. Lai. 1992. Class-based N-gram Models of Natural Language. *Comput. Linguist.*, 18(4):467–479. 273, 288, 290, 291

C. Buckley. 1994. Automatic query expansion using smart: Trec 3. In *Proc. of The third Text REtrieval Conference (TREC-3*, pp. 69–80. 144

S. Büttcher, C. Clarke, and G. V. Cormack. 2010. *Information Retrieval: Implementing and Evaluating Search Engines*. The MIT Press. xvii, 18, 165

F. Cacheda, V. Carneiro, D. Fernández, and V. Formoso. 2011. Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Trans. Web*, 5(1):2:1–2:33. DOI: 10.1145/1921591.1921593. 235

C. Campbell and Y. Ying. 2011. *Learning with Support Vector Machines*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers. DOI: 10.2200/S00324ED1V01Y201102AIM010. 311

J. Carbonell and J. Goldstein. 1998. The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. In *Proc. of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98,ACM, pp. 335–336, New York. DOI: doi=10.1.1.188.3982 321, 327

C.-C. Chang and C.-J. Lin. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27. 58

J. Chang, S. Gerrish, C. Wang, J. L. Boyd-graber, and D. M. Blei. 2009. Reading Tea Leaves: How Humans Interpret Topic Models. In Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, Curran Associates, Inc. 22, pp. 288–296. 272, 383, 384, 385, 410

K. W. Church and P. Hanks. 1990. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29. http://dl.acm.org/citation.cfm?id=89086.89095. 273

T. Cover and J. Thomas. 1991. *Elements of Information Theory*. New York: Wiley. DOI: DOI: 10.1002/047174882X 37, 473

B. Croft, D. Metzler, and T. Strohman. 2009. *Search Engines: Information Retrieval in Practice*, 1st ed., Addison-Wesley Publishing Company. xvii, 18, 165

D. Das and A. F. T. Martins. 2007. A Survey on Automatic Text Summarization. Technical report, Literature Survey for the Language and Statistics II course at Carnegie Mellon University. 318, 321, 327

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *J. Mach. Learn. Res.*, 9:1871–1874. 58

H. Fang, T. Tao, and C. Zhai. 2004. A formal study of information retrieval heuristics. In *Proc. of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, ACM, pp. 49–56, New York. DOI: 10.1145/1008992.1009004. 129

H. Fang, T. Tao, and C. Zhai. April 2011. Diagnostic evaluation of information retrieval models. *ACM Trans. Inf. Syst.*, 29(2):7:1–7:42. DOI: 10.1145/1961209.1961210. 88, 90, 129

R. Feldman and J. Sanger. 2007. *The Text Mining Handbook - Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press. 18

E. A. Fox, M. A. Gonçalves, and R. Shen. 2012. *Theoretical Foundations for Digital Libraries: The 5S (Societies, Scenarios, Spaces, Structures, Streams) Approach*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers. DOI: 10.2200/S00434ED1V01Y201207ICR022. 80

W. B. Frakes and R. A. Baeza-Yates, editors. 1992. *Information Retrieval: Data Structures & Algorithms*. Prentice-Hall, 18

K. Ganesan, C. Zhai, and J. Han. 2010. Opinosis: A graph-based approach to abstractive summarization of highly redundant opinions. In *Proc. of the 23rd International Conference on Computational Linguistics*, COLING '10, Association for Computational Linguistics, pp. 340–348, Stroudsburg, PA. 327

K. Ganesan, C. Zhai, and E. Viegas. 2012. Micropinion generation: an unsupervised approach to generating ultra-concise summaries of opinions. In *Proc. of the 21st World Wide Web Conference 2012, WWW 2012*, Lyon, France, April 16-20, 2012, pages 869–878. DOI: 10.1145/2187836.2187954 323

J. Gantz, and D. Reinsel. 2012. The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East, IDC Report, December, 2012. 3

A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. 1995. *Bayesian Data Analysis*. Chapman & Hall. 37

S. Ghemawat, H. Gobioff, and S.-T. Leung. 2003. *The Google file system*. In *Proc. of the nineteenth ACM symposium on Operating systems principles* (SOSP '03). ACM, New York, 29–43. 195

M. A. Gonçalves, E. A. Fox, L. T. Watson, and N. A. Kipp. 2004. Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries. *ACM Trans. Inf. Syst.*, 22(2):270–312. DOI: 10.1145/984321.984325. 84

D. A. Grossman and O. Frieder. Kluwer, 2004. *Information Retrieval - Algorithms and Heuristics, Second Edition*, vol. 15 of *The Kluwer International Series on Information Retrieval*. DOI: 10.1007/978-1-4020-3005-5. 18

G. Hamerly and C. Elkan. 2003. Learning the k in k-means. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS December 8-13, 2003*, Vancouver and Whistler, British Columbia, Canada], pp. 281–288. DOI: doi=10.1.1.9.3574 295

J. Han. 2005. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA. 296

D. Harman. 2011. *Information Retrieval Evaluation*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers. DOI: 10.1145/215206.215351 168, 188

M. A. Hearst. 2009. *Search User Interfaces*. 1st ed. Cambridge University Press, New York. 19, 85

J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. 2004. Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. Inf. Syst.*, 22(1):5–53. DOI: 10.1145/963770.963772 235

J. L. Hodges and E. L. Lehmann. 1970. *Basic Concepts of Probability and Statistics*. Holden Day, San Francisco. 36

T. Hofmann. 1999. Probabilistic Latent Semantic Analysis. In *Proc. of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, Morgan Kaufmann Publishers Inc., pp. 289–296, San Francisco, CA. DOI: 10.1145/312624.312649 370, 385

A. Huang. 2008. Similarity Measures for Text Document Clustering. In *Proc. of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008)*, Christchurch, New Zealand, pages 49–56. 280

F. Jelinek. 1997. *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, MA. 30, 54

J. Jiang. 2012. Information extraction from text, In Charu C. Aggarwal and ChengXiang Zhai (Eds.), Mining Text Data, Springer, pp. 11–41. 19, 55

S. Jiang and C. Zhai. 2014. Random walks on adjacency graphs for mining lexical relations from big text data. In *2014 IEEE International Conference on Big Data, Big Data 2014*, Washington, DC, USA, October 27-30, pages 549–554. DOI: 10.1109/BigData.2014.7004272. 273

Y. Jo and A. H. Oh. 2011. Aspect and sentiment unification model for online review analysis. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, ACM, pp. 815–824, New York. DOI: 10.1145/1935826.1935932. 410

T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Trans. Inf. Syst.*, 25(2). DOI: 10.1145/1229179.1229181. 144

D. Jurafsky and J. H. Martin. 2009. *Speech and Language Processing*. 2nd ed. Prentice-Hall, Inc., Upper Saddle River, NJ. 19, 54

D. Kelly. 2009. Methods for Evaluating Interactive Information Retrieval Systems with Users. *Foundations and Trends in Information Retrieval*, 3(1-2):1–224. DOI: 10.1561/1500000012 168, 188

D. Kelly and J. Teevan. 2003. Implicit feedback for inferring user preference: A bibliography. *SIGIR Forum*, 37(2):18–28. DOI: 10.1145/959258.959260. 144

H. D. Kim, M. Castellanos, M. Hsu, C. Zhai, T. Rietz, and D. Diermeier. 2013. Mining causal topics in text data: iterative topic modeling with time series feedback. In *Proc. of the 22nd ACM international conference on Conference on information and knowledge management*, CIKM '13, ACM pages 885–890, New York, NY. DOI: 10.1145/2505515.2505612. 435, 438, 439, 440

J. M. Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632. DOI: 10.1145/324133.324140. 216

J. M. Kleinberg. 2002. An impossibility theorem for clustering. In *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002*, December 9-14, 2002, Vancouver, British Columbia, Canada], pp. 446–453. http://papers.nips.cc/paper/2340-an-impossibility-theorem-for-clustering. 296

D. Koller and N. Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press. 385

J. Lafferty and C. Zhai. 2003. Probabilistic relevance models based on document and query generation. In W. Bruce Croft and John Lafferty, editors, *Language Modeling and Information Retrieval*. Kluwer Academic Publishers. DOI: 10.1007/978-94-017-0171-6_1 87, 113

D. Lin. 1999. Automatic identification of non-compositional phrases. In *Proc. of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, Association for Computational Linguistics, pages 317–324, Stroudsburg, PA. DOI: 10.3115/1034678.1034730. 273, 291

J.Lin and C. Dyer. 2010. *Data-Intensive Text Processing with MapReduce*. Morgan and Claypool Publishers. DOI: 10.2200/S00274ED1V01Y201006HLT007. 198, 216

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers. DOI: 10.2200/S00416ED1V01Y201204HLT016. 410

T.-Y. Liu. 2009. Learning to rank for information retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331. DOI: 10.1561/1500000016. 216

Y. Lv and C. Zhai. 2009. A comparative study of methods for estimating query language models with pseudo feedback. In *Proc. of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, ACM, pp. 1895–1898, New York. DOI: 10.1145/1645953.1646259. 144

Y. Lv and C. Zhai. 2010. Positional relevance model for pseudo-relevance feedback. In *Proc. of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, ACM, pages 579–586, New York. DOI: 10.1145/1835449.1835546. 144

Y. Lv and C. Zhai. 2011. Lower-bounding Term Frequency Normalization. In *Proc. of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pp. 7–16. DOI: 10.1145/2063576.2063584 88, 110

P. Lyman, H. R. Varian, K. Swearingen, P. Charles, N. Good, L.L. Jordan, and J. Pal. 2003. How much information? http://www2.sims.berkeley.edu/research/projects/how-much-info-2003. 3

C. D. Manning and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA. 19, 54, 273

C. D. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York. xvii, 18, 165, 315

M. E. Maron and J. L. Kuhns. 1960. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM*, 7:216–244. DOI: 10.1145/321033.321035 87

S. Massung and C. Zhai. 2015. SyntacticDiff: Operator-Based Transformation for Comparative Text Mining. In *Proc. of the 3rd IEEE International Conference on Big Data*, pp. 571–580. 306

S. Massung and C. Zhai. 2016. Non-Native Text Analysis: A Survey. *The Journal of Natural Language Engineering*, 22(2):163–186. DOI: 10.1017/S1351324915000303 306

S. Massung, C. Zhai, and J.Hockenmaier. 2013. Structural Parse Tree Features for Text Representation. In *IEEE Seventh International Conference on Semantic Computing*, pp. 9–13. DOI: 10.1109/ICSC.2013.13 305

J. D. McAuliffe and D. M. Blei. 2008. Supervised topic models. In J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis, eds., *Advances in Neural Information Processing Systems 20*, pages 121–128. Curran Associates, Inc. 386

G. J. McLachlan and T. Krishnan. 2008. *The EM algorithm and extensions*. 2nd ed. Wiley Series in Probability and Statistics. Hoboken, NJ., Wiley. http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+52983362X&sourceid=fbw_bibsonomy. DOI: 10.1002/9780470191613 466

Q. Mei. 2009. Contextual text mining. Ph.D. Dissertation, University of Illinois at Urbana-Champaign. 440

Q. Mei and C. Zhai. 2006. A mixture model for contextual text mining. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, ACM, pp. 649–655, New York. DOI: 10.1145/1150402.1150482. 423, 440

Q. Mei, D. Xin, H. Cheng, J. Han, and C. Zhai. 2006. Generating semantic annotations for frequent patterns with context analysis. In *Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, ACM, pp. 337–346, New York. DOI: 10.1145/1150402.1150441. 417

Q. Mei, C. Liu, H. Su, and C. Zhai. 2006. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *Proc.of the 15th international conference on World Wide Web (WWW '06)*. ACM. New York, 533–542. DOI: 10.1145/1135777.1135857. 425, 426

Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. 2007a. Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proc. of the 16th International Conference on World Wide Web*, WWW '07, ACM, pp. 171–180, New York. DOI: 10.1145/1242572.1242596. 410

Q. Mei, X. Shen, and C. Zhai. 2007b. Automatic labeling of multinomial topic models. In *Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Jose, California, August 12-15, 2007, pp. 490–499. DOI: 10.1145/1281192.1281246. 278

Q. Mei, D. Cai, D. Zhang, and C. Zhai. 2008. Topic modeling with network regularization. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, ACM, pp. 101–110, New York. DOI: 10.1145/1367497.1367512. 431, 432, 440

T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association*, Makuhari, Chiba, Japan, September 26-30, 2010, pp. 1045–1048. http://www.isca-speech.org/archive/interspeech_2010/i10_1045.html. 292

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, NV, pp. 3111–3119. 273, 292, 293

T. M. Mitchell. 1997. *Machine learning*. McGraw Hill Series in Computer Science. McGraw-Hill. 19, 37, 315

M.-F. Moens. 2006. *Information Extraction: Algorithms and Prospects in a Retrieval Context (The Information Retrieval Series)*. Springer-Verlag New York, Inc., Secaucus, NJ. DOI: 10.1007/978-1-4020-4993-4. 55

I. J. Myung. 2003. Tutorial on maximum likelihood estimation. *J. Math. Psychol.*, 47(1):90–100. DOI: 10.1016/S0022-2496(02)00028-7. 36

A. Nenkova and K. McKeown. 2012. A survey of text summarization techniques. In Charu C. Aggarwal and C. Zhai, eds, *Mining Text Data*, pp. 43–76. Springer US. DOI: 10.1007/978-1-4614-3223-4_3. 327

L. Page, S. Brin, R. Motwani, and T. Winograd. 1999. The PageRank Citation Ranking: Bringing Order to the Web. http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf. 216

B. Pang and L. Lee. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135. DOI: 10.1561/1500000011 409, 410

J. M. Ponte and W. B. Croft. 1998. A language modeling approach to information retrieval. In *Proc. of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, ACM, pp. 275–281, New York, NY. DOI: 10.1145/290941.291008. 87, 90, 128, 427

J. R. Quinlan. 1986. Induction of Decision Trees. *Machine Learning*, 1(1):81–106. DOI: 10.1007/BF00116251. 301

D. R. Radev, H. Jing, M. Styś, and D. Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938. DOI: 10.1016/j.ipm.2003.10.006. 327

D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. 2009. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, Association for Computational Linguistics, pages 248–256, Stroudsburg, PA. 386

E. Reiter and R. Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, New York. 324, 327

F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. 2010. *Recommender Systems Handbook*. 1st ed. Springer-Verlag New York, Inc. DOI: 10.1007/978-0-387-85820-3 235

C. J. Van Rijsbergen. 1979. *Information Retrieval*. 2nd ed. Butterworth-Heinemann, Newton, MA.

S. Robertson and K. Sparck Jones. 1976. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146. 87

S. E. Robertson. 1997. Readings in Information Retrieval. In *The Probability Ranking Principle in IR*, San Francisco, CA, Morgan Kaufmann Publishers Inc. pp. 281–286. 84, 85

S. Robertson and H. Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.*, 3(4):333–389. DOI: 10.1561/1500000019. 88, 89, 129

S. Robertson, H. Zaragoza, and M. Taylor. 2004. Simple BM25 Extension to Multiple Weighted Fields. In *Proc. of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, pp. 42–49. DOI: 10.1145/1031171.1031181 110

C. Roe. 2012. The growth of unstructured data: what to do with all those zettabytes? http://www.dataversity.net/the-growth-of-unstructured-data-what-are-we-going-to-do-with-all-those-zettabytes/. 3

R. Rosenfeld. 2000. Two decades of statistical language modeling: Where do we go from here. In *Proceedings of the IEEE*. 54

G. Salton. 1989. *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*. Addison-Wesley. 18

G. Salton and M. McGill. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill. 18

G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620. 87

G. Salton and C. Buckley. 1990. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41:288–297. 144

M. Sanderson. 2010. Test Collection Based Evaluation of Information Retrieval Systems. *Foundations and Trends in Information Retrieval*, 4(4):247–375. 168, 188

M. Sanderson and W. B. Croft. 2012. The history of information retrieval research. *Proc. of the IEEE*, 100(Centennial-Issue):1444–1451, 2012. DOI: 10.1109/JPROC.2012.2189916. 85

S. Sarawagi. 2008. Information extraction. *Found. Trends databases*, 1(3):261–377. DOI: 10.1561/1900000003. 19, 55

F. Sebastiani. 2002. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47. DOI: 10.1145/505282.505283. 315

G. Shani and A. Gunawardana. 2011. Evaluating Recommendation Systems. In *Recommender Systems Handbook*, 2nd ed., pp. 257–297. Springer, New York, NY. DOI: 10.1007/978-0-387-85820-3_8. 235

F. Silvestri. 2010. Mining query logs: Turning search usage data into knowledge. *Found. Trends Inf. Retr.*, 4:1–174. DOI: 10.1561/1500000013 144

A. Singhal, C. Buckley, and Mandar Mitra. 1996. Pivoted document length normalization. In *Proc. of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96,ACM, pp. 21–29, New York. DOI: 10.1145/243199.243206. 89, 106, 128

N. Smith. 2010. Text-driven forecasting. http://www.cs.cmu.edu/ñasmith/papers/smith.whitepaper10.pdf. 440

Mark D. Smucker, James Allan, and Ben Carterette. 2007. A Comparison of Statistical Significance Tests for Information Retrieval Evaluation. In *Proc. of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, ACM, pp. 623–632, New York. DOI: 10.1145/1321440.1321528. 185

K. Sparck Jones and P. Willett, eds. 1997. *Readings in Information Retrieval*. San Francisco, CA, Morgan Kaufmann Publishers Inc. 18, 85, 188

N. Spirin and J. Han. May 2012. Survey on Web Spam Detection: Principles and Algorithms. *SIGKDD Explor. Newsl.*, 13(2):50–64. DOI: 10.1145/2207243.2207252. 191

E. Stamatatos. 2009. A Survey of Modern Authorship Attribution Methods. *J. Am. Soc. Inf. Sci. Technol.*, 60(3):538–556. DOI: 10.1002/asi.v60:3 305

M. Steinbach, G. Karypis, and V. Kumar. 2000. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*. 296

J. Steinberger and K. Jezek. 2009. Evaluation measures for text summarization. *Computing and Informatics*, 28(2):251–275. 327

M. Steyvers and T. Griffiths. 2007. Probabilistic topic models. *Handbook of Latent Semantic Analysis*, 427(7):424–440. 385, 386

Y. Sun and J. Han. 2012. *Mining Heterogeneous Information Networks: Principles and Methodologies*. Morgan & Claypool Publishers. DOI: 10.2200/S00433ED1V01Y201207DMK005. 440

I. Titov and R. McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proc. of the 17th International Conference on World Wide Web*, WWW '08, ACM, pp. 111–120, New York. DOI: 10.1145/1367497.1367513. 410

H. Turtle and W. B. Croft. 1990. Inference networks for document retrieval. In *Proc. of the 13th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '90, ACM, pp. 1–24, New York. DOI: 10.1145/96749.98006. 88

Princeton University. 2010. About wordnet. http://wordnet.princeton.edu. 395

C. J. van Rijsbergen. 1979. *Information Retrieval*. Butterworths. 18

H. Wang, Yue Lu, and C. Zhai. 2010. Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach. In *Proc. of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, ACM, pp. 783–792, New York. DOI: 10.1145/1835804.1835903. 318, 327, 405, 406, 407, 408, 409, 410

H. Wang, Y. Lu, and C. Zhai. 2011. Latent Aspect Rating Analysis Without Aspect Keyword Supervision. In *Proc. of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, ACM, pp. 618–626, New York. DOI: 10.1145/2020408.2020505. 318, 327, 405, 410

J. Weizenbaum. 1966. ELIZA—A Computer Program for the Study of Natural Language Communication Between Man and Machine, Communications of the ACM 9 (1): 36–45, DOI: 10.1145/265153.365168. 44

J. S. Whissell and C. L. A. Clarke. 2013. Effective Measures for Inter-document Similarity. In *Proc. of the 22nd ACM International Conference on Conference on Information & Knowledge Management*, CIKM '13, ACM, pages 1361–1370, New York. DOI: 10.1145/2505515.2505526. 279

R. W. White and R. A. Roth. 2009. *Exploratory Search: Beyond the Query-Response Paradigm*. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers. DOI: 10.2200/S00174ED1V01Y200901ICR003. 85

R. W. White, B. Kules, S. M. Drucker, and m.c. schraefel. 2006. Introduction. *Commun. ACM*, 49(4):36–39. DOI: 10.1145/1121949.1121978. 85

I. H. Witten, A. Moffat, and T. C. Bell. 1999. *Managing Gigabytes (2Nd Ed.): Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishers Inc., San Francisco, CA. 18, 165

C.F J. Wu. 1983. On the convergence properties of the EM algorithm. *Ann. of stat.*, 95–103. 368

J. Xu and W. B. Croft. 1996. Query expansion using local and global document analysis. In *Proc. of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '96, ACM, pp. 4–11, New York. DOI: 10.1145/243199.243202. 144

Y. Yang. 1999. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1:67–88. 315

C. Zhai. 1997. Exploiting context to identify lexical atoms—a statistical view of linguistic context. In *Proc. of the International and Interdisciplinary Conference on Modelling and Using Context (CONTEXT-97)*, pages 119–129. Rio de Janeiro, Brazil. 273, 291

C. Zhai. 2008. *Statistical Language Models for Information Retrieval*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers. DOI: 10.2200/S00158ED1V01Y200811HLT001. 55, 87, 128, 129

C. Zhai and J. Lafferty. 2001. Model-based Feedback in the Language Modeling Approach to Information Retrieval. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, ACM, pp. 403–410, New York. DOI: 10.1145/502585.502654. 143, 466, 473

C. Zhai and J. Lafferty. 2004. A Study of Smoothing Methods for Language Models Applied to Information Retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214. 475

C. Zhai, P. Jansen, E. Stoica, N. Grot, and D. A. Evans. 1998. Threshold Calibration in CLARIT Adaptive Filtering. In *Proc. of Seventh Text REtrieval Conference (TREC-7)*, pp. 149–156. 227

C. Zhai, P. Jansen, and D. A. Evans. 2000. Exploration of a heuristic approach to threshold learning in adaptive filtering. In *SIGIR*, ACM, pp. 360–362. DOI: 10.1145/345508.345652. 235

C. Zhai, A. Velivelli, and B. Yu. 2004. A cross-collection mixture model for comparative text mining. In *Proc. of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, ACM, pp. 743–748, New York. DOI: 10.1145/1014052.1014150. 423

D. Zhang, C. Zhai, J. Han, A. Srivastava, and N. Oza. 2009. Topic modeling for OLAP on multidimensional text databases: topic cube and its applications. *Stat. Anal. Data Min.* 2, 5–6 (December 2009), 378–395. DOI: 10.1002/sam.v2.5/6. 440

J. Zhu, A. Ahmed, and E. P. Xing. 2009. Medlda: Maximum margin supervised topic models for regression and classification. In *Proc. of the 26th Annual International Conference on Machine Learning*, ICML '09, ACM, pp. 1257–1264, New York. DOI: 10.1145/1553374.1553535. 386

G. K. Zipf. 1949. *Human Behavior and the Principle of Least-Effort*. Cambridge, MA, Addison-Wesley, 162

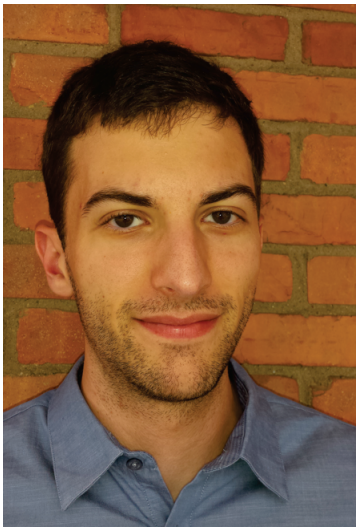
Index

Authors' Biographies

ChengXiang Zhai

ChengXiang Zhai is a Professor of Computer Science and Willett Faculty Scholar at the University of Illinois at Urbana–Champaign, where he is also affiliated with the Graduate School of Library and Information Science, Institute for Genomic Biology, and Department of Statistics. He received a Ph.D. in Computer Science from Nanjing University in 1990, and a Ph.D. in Language and Information Technologies from Carnegie Mellon University in 2002. He worked at Clairvoyance Corp. as a Research Scientist and then Senior Research Scientist from 1997–2000. His research interests include information retrieval, text mining, natural language processing, machine learning, biomedical and health informatics, and intelligent education information systems. He has published over 200 research papers in major conferences and journals. He served as an Associate Editor for *Information Processing and Management*, as an Associate Editor of *ACM Transactions on Information Systems*, and on the editorial board of *Information Retrieval Journal*. He was a conference program co-chair of ACM CIKM 2004, NAACL HLT 2007, ACM SIGIR 2009, ECIR 2014, ICTIR 2015, and WWW 2015, and conference general co-chair for ACM CIKM 2016. He is an ACM Distinguished Scientist and a recipient of multiple awards, including the ACM SIGIR 2004 Best Paper Award, the ACM SIGIR 2014 Test of Time Paper Award, Alfred P. Sloan Research Fellowship, IBM Faculty Award, HP Innovation Research Program Award, Microsoft Beyond Search Research Award, and the Presidential Early Career Award for Scientists and Engineers (PECASE).

Sean Massung



Sean Massung is a Ph.D. candidate in computer science at the University of Illinois at Urbana–Champaign, where he also received both his B.S. and M.S. degrees. He is a co-founder of MeTA and uses it in all of his research. He has been instructor for CS 225: Data Structures and Programming Principles, CS 410: Text Information Systems, and CS 591txt: Text Mining Seminar. He is included in the 2014 List of Teachers Ranked as Excellent at the University of Illinois and has received an Outstanding Teaching Assistant Award and CS@Illinois Outstanding Research Project Award. He has given talks at Jump Labs Champaign and at UIUC for Data and Information Systems Seminar, Intro to Big Data, and Teaching Assistant Seminar. His research interests include text mining applications in information retrieval, natural language processing, and education.