LANGUAGE MODELS

Djoerd Hiemstra
University of Twente
http://www.cs.utwente.nl/~hiemstra

SYNONYMS

GENERATIVE MODELS

DEFINITION

A language model assigns a probability to a piece of unseen text, based on some training data. For example, a language model based on a big English newspaper archive is expected to assign a higher probability to "a bit of text" than to "aw pit tov tags", because the words in the former phrase (or word pairs or word triples if so-called *N*-GRAM MODELS are used) occur more frequently in the data than the words in the latter phrase. For information retrieval, typical usage is to build a language model for each document. At search time, the top ranked document is the one which' language model assigns the highest probability to the query.

HISTORICAL BACKGROUND

The term *language models* originates from probabilistic models of language generation developed for automatic speech recognition systems in the early 1980's [9]. Speech recognition systems use a language model to complement the results of the *acoustic model* which models the relation between words (or parts of words called phonemes) and the acoustic signal. The history of language models, however, goes back to beginning of the 20th century when Andrei Markov used language models (Markov models) to model letter sequences in works of Russian literature [3]. Another famous application of language models are Claude Shannon's models of letter sequences and word sequences, which he used to illustrate the implications of coding and information theory [17]. In the 1990's language models were applied as a general tool for several natural language processing applications, such as part-of-speech tagging, machine translation, and optical character recognition. Language models were applied to information retrieval by a number of research groups in the late 1990's [4, 7, 14, 15]. They became rapidly popular in information retrieval research. By 2001, the ACM SIGIR conference had two separate sessions on language models containing 5 papers in total [13]. In 2003, a group of leading information retrieval researchers published a research roadmap "Challenges in Information Retrieval and Language Modeling" [1], indicating that the future of information retrieval and the future of language modeling can not be seen apart from each other.

SCIENTIFIC FUNDAMENTALS

Language models are generative models, i.e., models that define a probability mechanism for generating language. Such generative models might be explained by the following probability mechanism: Imagine picking a term T at random from this page by pointing at the page with closed eyes. This mechanism defines a probability $P(T|D)$, which could be defined as the relative frequency of the occurrence of the event, i.e., by the number of occurrences of a word on the page divided by the total number of terms on the page. Suppose the process is repeated n times, picking one at a time the terms T_1, T_2,..., T_n. Then, assuming independence between the successive events, the probability of the terms given the document D is defined as follows:

$$P(T_1, T_2, \cdots, T_n|D) \;=\; \prod_{i=1}^{n} P(T_i|D) \tag{1}$$

A simple language modeling approach would compute Equation 1 for each document in the collection, and rank the documents accordingly. A potential problem might be the following: The equation will assign zero probability to a sequence of terms unless all terms occur in the document. So, a language modeling system that uses Equation 1 will not retrieve a document unless it contains all query terms. This might be reasonable for a web search engine that typically processes small queries to search a vast amount of data, but for many other information retrieval applications this behavior is a problem. A standard solution is to use *linear interpolation smoothing* (see PROBABILITY SMOOTHING) of the document model $P(T|D)$ with a collection model $P(T|C)$, which is defined as follows.

$$P(T_1, T_2, \cdots, T_n|D) \;=\; \prod_{i=1}^{n} \Big(\lambda P(T_i|D) + (1{-}\lambda) P(T_i|C)\Big) \tag{2}$$

This way, a term that does not occur in the document will not be assigned zero probability but instead a probability proportional to its number of occurrences in the entire collection C. Here, λ is an unknown probability that should be tuned to optimize retrieval effectiveness. Linear interpolation smoothing was used in several early language modeling approaches [7, 14].

Implementation Although the language modeling equations above suggest the need to compute probabilities for all documents in the collection, this is unnecessary in practice. In fact, most language modeling approaches can be implemented efficiently by the use of standard inverted index search systems. This can be seen by the equation below which can be derived from Equation 2 by two basic transformations: First, dividing it by the probability of the collection ground model; and second, taking the logarithm.

$$P(T_1, T_2, \cdots, T_n|D) \;\propto\; \sum_{i=1}^{n} \log\Big(1 + \frac{\lambda P(T_i|D)}{(1{-}\lambda) P(T_i|C)}\Big) \tag{3}$$

Equation 3 does no longer produce probabilities, but it ranks the documents in the exact same order as Equation 2, because the collection model does not depend on the document, and the logarithm is a strictly monotonic function. Taking the logarithm prevents the implementation from running out of the precision of its (floating point) representation of probabilities, which can become very small because the probabilities are multiplied for every query term. Similar to for instance vector space models in information retrieval, ranking is defined by a simple sum of term weights, for which terms that do not match a document get a zero weight. Interestingly, the resulting "term weight" can be seen as a variant of *tf.idf* weights, which are often used in vector space models.

Document priors The equations above define the probability of a query given a document, but obviously, the system should rank by the probability of the documents given the query. These two probabilities are related by Bayes' rule as follows.

$$P(D|T_1, T_2, \cdots, T_n) \;=\; \frac{P(T_1, T_2, \cdots, T_n|D) P(D)}{P(T_1, T_2, \cdots, T_n)} \tag{4}$$

The left-hand side of Equation 4 cannot be used directly because the independence assumption presented above assumes term independence given the document. So, in order to compute the probability of the document D given the query, Equation 2 need to be multiplied by $P(D)$ and divided by $P(T_1, \cdots, T_n)$. Again, as stated in the previous paragraph, the probabilities themselves are of no interest, only the ranking of the document by the probabilities is. And since $P(T_1, \cdots, T_n)$ does not depend on the document, ranking the documents by the numerator of the right-hand side of Equation 4 will rank them by the probability given the query. This shows the importance of $P(D)$: The marginal probability, or *prior probability* of the document, i.e., it is the probability that the document is relevant if the query is ignored. For instance, we might assume that long documents are more likely to be usefull than short documents [7, 14]. In web search, such a so-called static ranking (a ranking that is independent of the query) is commonly used. For instance, documents with many links pointing to them are more likely to be relevant, or documents with short URLs are more likely to be relevant. The prior probability of a document is a powerful way to incorporate static ranking in the language modeling approach [11].

Document generation models An implicit assumption of the language models presented until now is that there is more information available about the documents than about the query. In some applications however, the situation is reversed. For instance in *topic tracking*, a system has the task to track a stream of chronologically ordered stories. For each story in the stream, the system has to decide if it is on topic, or not. The target topic is usually based on a number of example stories on a certain topic, there is more information available about the topic than about a single story. Unlike query generation models, document generation models need some form of normalization because documents will have different lengths: The probability of generating a document tends to be smaller for long documents than for short documents. Therefore, several normalization techniques might be applied, such as normalization by document length and additional Gaussian normalization [10, 18]. *Relevance feedback*, i.e, the user marked some documents as relevant, is another situation in which there is more knowledge available about the query than about each single document. If some relevant documents are known, or if the top ranked documents are assumed to be relevant, then those documents might be used to generate a new, improved query [20]. As an example, consider the following so-called *relevance models* approach [12]

$$P(Q|T_1, \cdots, T_n) \; \propto \; \sum_d \Big(P(D\!=\!d) P(Q|D\!=\!d) \prod_{i=1}^{n} P(T_i\!=\!t_i|D\!=\!d) \Big) \tag{5}$$

Here, the formula defines the probability of a new word Q, given the original query $T_1, \cdots, T_n$ by marginalizing over all documents. In practice, only the top ranked documents for the query $T_1, \cdots$ are used. Interestingly, the relevance model might be used to infer other information from the top ranked documents, for instance the person that is most often mentioned for a certain query, so-called *expert search* [2].

Translation models Language models for information retrieval are generative models, and therefore easily combined with other generative models. To add a model of term translation, the following probability mechanism applies: Imagine picking an English term T at random from this page by pointing at the page with closed eyes (which defines a probability $P(T|D)$), and then translate the term T by picking from the term's entry in a English-Dutch dictionary at random a Dutch term S (with probability $P(S|T)$). The model might be used in a cross-language retrieval system to rank English documents given a Dutch query $S_1, \cdots, S_n$ by the following probability [4, 6, 12, 19]:

$$P(S_1, S_2, \cdots, S_n|D) \; = \; \prod_{i=1}^{n} \sum_t \Big(P(S_i\!=\!s_i|T_i\!=\!t) (\; \lambda P(T_i\!=\!t|D) \; + \; (1\!-\!\lambda) P(T_i\!=\!t|C) \;) \Big) \tag{6}$$

Here, Dutch is the source language and English the target language. The formula uses linear interpolation smoothing of the document model with the target language background model $P(T|C)$ (English in the example) at the right-hand side of the formula. In some formulations, the translation model is smoothed with the source language background model $P(S|C)$ which a estimated on auxiliary data. The two background models are related as follows: $P(S|C) = \sum_t P(S|T\!=\!t)P(T\!=\!t|C)$. The translation probabilities are often estimated from parallel corpora, i.e., from texts in the target language and its translations in the source language [6, 19]. Translation models might also be used in a monolingual setting to account for synonyms and other related words [4].

Aspect models In *aspect models*, also called *probabilistic latent semantic indexing* models, documents are modeled as mixtures of aspect language models. In terms of a generative model it can be defined in the following way [8]: 1) select a document D with probability $P(D)$, 2) pick a latent aspect Z with probability $P(Z|D)$, 3) generate a term T with probability $P(T|Z)$ independent of the document, 4) repeat Step 2 and Step 3 until the desired number of terms is reached. This leads to Equation 7.

$$P(T_1, T_2, \cdots, T_n|D) \; \propto \; \prod_{i=1}^{n} \Big(\sum_z (P(T_i|Z\!=\!z) P(Z\!=\!z|D)) \Big) \tag{7}$$

The aspects might correspond with the topics or categories of documents in the collection such as “health”, “family”, “Hollywood”, etc. The aspect Z is a hidden, unobserved variable, so probabilities concerning Z cannot be estimated from direct observations. Instead, the Expectation Maximization (EM) algorithm can be applied [9]. The algorithm starts out with a random initialization of the probabilities, and then iteratively re-estimates the probabilities to arrive at a local maximum of the likelihood function. It has been shown that the EM algorithm is sensitive to the initialization, and an unlucky initialization results in a non-optimal local maximum. As a solution, clustering of documents has been proposed to initialize the models [16]. Another alternative is latent semantic Dirichlet allocation [5] which has less free parameters, and therefore is less sensitive to the initialization.

KEY APPLICATIONS

This entry focuses on the application of language models to information retrieval. The applications presented include newswire and newspaper search [4, 5, 15], web search [11], cross-language search [6, 19], topic detection and tracking [10, 18], and expert search [2]. However, language models have been used in virtually every application that needs processing of natural language texts, including automatic speech recognition, part-of-speech tagging, machine translation, and optical character recognition.

CROSS REFERENCE

PROBABILITY SMOOTHING, N-GRAM MODELS

RECOMMENDED READING

[1] James Allan (editor), Jay Aslam, Nicholas Belkin, Chris Buckley, Jamie Callan, Bruce Croft (editor), Sue Dumais, Norbert Fuhr, Donna Harman, David J. Harper, Djoerd Hiemstra, Thomas Hofmann, Eduard Hovy, Wessel Kraaij, John Lafferty, Victor Lavrenko, David Lewis, Liz Liddy, R. Manmatha, Andrew McCallum, Jay Ponte, John Prager, Dragomir Radev, Philip Resnik, Stephen Robertson, Roni Rosenfeld, Salim Roukos, Mark Sanderson, Richard Schwartz, Amit Singhal, Alan Smeaton, Howard Turtle, Ellen Voorhees, Ralph Weischedel, Jinxi Xu, and ChengXiang Zhai. Challenges in Information Retrieval and Language Modeling. *SIGIR Forum 37*(1), 2003

[2] Kristian Balog, Leif Azzopardi, and Maarten de Rijke. Formal models for expert finding in enterprise corpora. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'06)*, pages 43–50, 2006.

[3] Gely P. Basharin, Amy N. Langville, and Valeriy A. Naumov. The life and work of A.A. Markov. *Linear Algebra and its Applications 386*, pages 3–26, 2004.

[4] Adam Berger and John Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, pages 222–229, 1999.

[5] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research 3*(5):993–1022, 2003.

[6] Djoerd Hiemstra and Franciska de Jong. Disambiguation strategies for cross-language information retrieval. *Lecture Notes in Computer Science volume 1696: Proceedings of the European Conference on Digital Libraries, Springer-Verlag*, pages 274–293, 1999.

[7] Djoerd Hiemstra and Wessel Kraaij. Twenty-One at TREC-7: Ad-hoc and cross-language track. In *Proceedings of the seventh Text Retrieval Conference TREC-7*, pages 227–238. NIST Special Publication 500-242, 1998.

[8] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'99)*, pages 50–57, 1999.

[9] Frederick Jelinek. *Statistical Methods for Speech Recognition.* MIT Press, 1997.

[10] Hubert Jin, Rich Schwartz, Sreenivasa Sista, Frederick Walls. Topic Tracking for Radio, TV Broadcast and Newswire. In *Proceedings of the DARPA Broadcast News Workshop*, 1999.

[11] Wessel Kraaij, Thijs Westerveld, and Djoerd Hiemstra. The importance of prior probabilities for entry page search. In *Proceedings of the 25th ACM Conference on Research and Development in Information Retrieval (SIGIR'02)*, pages 27–34 2002.

[12] Victor Lavrenko and W. Bruce Croft. Relevance models in information retrieval. In W. Bruce Croft and John Lafferty (eds.) *Language Modeling for Information Retrieval, Kluwer Academic Publishers*, pages 11–56, 2003.

[13] Donald H. Kraft, W. Bruce Croft, David J. Harper, and Justin Zobel (eds.). Proceedings of the 24th ACM Conference on Research and Development in Information Retrieval (SIGIR'01). *Association for Computing Machinery*, 2001.

[14] David R.H. Miller, Timothy Leek, and Richard M. Schwartz. A hidden Markov model information retrieval system. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, pages 214–221, 1999.

[15] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st ACM Conference on Research and Development in Information Retrieval (SIGIR'98)*, pages 275–281, 1998.

[16] Richard M. Schwartz, Sreenivasa Sista, Timothy Leek. Unsupervised Topic Discovery. In *Proceedings of the Language Models for Information Retrieval workshop (LMIR)*, 2001.

[17] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[18] Martijn Spitters, Wessel Kraaij. Language models for topic tracking. In W. Bruce Croft and John Lafferty (eds.) *Language Modeling for Information Retrieval, Kluwer Academic Publishers*, pages 95–124, 2003.

[19] Jinxi Xu and Ralph Weischedel. A probabilistic approach to term translation for cross-lingual retrieval. In W. Bruce Croft and John Lafferty (eds.) *Language Modeling for Information Retrieval, Kluwer Academic Publishers*, pages 125–140, 2003.

[20] Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the Tenth ACM International Conference on Information and Knowledge Management (CIKM'01)*, pages 403–410, 2001.

PROBABILITY SMOOTHING

Djoerd Hiemstra
University of Twente
http://www.cs.utwente.nl/~hiemstra

DEFINITION

Probability smoothing is a language modeling technique that assigns some non-zero probability to events that were unseen in the training data. This has the effect that the probability mass is divided over more events, hence the probability distribution becomes more *smooth*.

MAIN TEXT

Smoothing overcomes the so-called *sparse data problem*, that is, many events that are plausible in reality are not found in the data used to estimate probabilities. When using maximum likelihood estimates, unseen events are assigned zero probability. In case of information retrieval, most events are unseen in the data, even if simple unigram language models are used (see N-GRAM MODELS): Documents are relatively short (say on average several hundreds of words), whereas the vocabulary is typically big (maybe millions of words), so the vast majority of words does not occur in the document. A small document about "information retrieval" might not mention the word "search", but that does not mean it is not relevant to the query "text search". The sparse data problem is the reason that it is hard for information retrieval systems to obtain high recall values without degrading values for precision, and smoothing is a means to increase recall (possibly degrading precision in the process). Many approaches to smoothing are proposed in the field of automatic speech recognition [1]. A smoothing method may be as simple so-called Laplace smoothing, which adds an extra count to every possible word. The following equations show respectively (8) the unsmoothed, or maximum likelihood estimate, (9) Laplace smoothing, (10) Linear interpolation smoothing, and (11) Dirichlet smoothing [3]:

$$P_{ML}(T\!=\!t|D\!=\!d) \;=\; tf(t,d) \;\;/\;\; \textstyle\sum_{t'} tf(t',d) \tag{8}$$

$$P_{LP}(T\!=\!t|D\!=\!d) \;=\; (tf(t,d)+1) \;\;/\;\; \textstyle\sum_{t'}(tf(t',d)+1) \tag{9}$$

$$P_{LI}(T\!=\!t|D\!=\!d) \;=\; \lambda P_{ML}(T\!=\!t|D\!=\!d) + (1\!-\!\lambda) P_{ML}(T\!=\!t|C) \tag{10}$$

$$P_{Di}(T\!=\!t|D\!=\!d) \;=\; (tf(t,d)+\mu P_{ML}(T\!=\!t|C)) \;\;/\;\; ((\textstyle\sum_{t'} tf(t',d))+\mu) \tag{11}$$

Here, $tf(t,d)$ is the frequency of occurrence of the term t in the document d, and $P_{ML}(T|C)$ is the probability of a term occurring in the entire collection C. Both linear interpolation smoothing (see also the entry LANGUAGE MODELS) and Dirichlet smoothing assign a probability proportional to the term occurrence in the collection to unseen terms. Here, λ $(0 < \lambda < 1)$ and μ $(\mu > 0)$ are unknown parameters that should be tuned to optimize retrieval effectiveness. Linear interpolation smoothing has the same effect on all documents, whereas Dirichlet smoothing has a relatively big effect on small documents, but a relatively small effect on bigger documents. Many smoothed estimators used for language models in information retrieval (including Laplace and Dirichlet smoothing) are approximations to the *Bayesian predictive distribution* [2].

CROSS REFERENCE

LANGUAGE MODELS, N-GRAM MODELS

RECOMMENDED READING

[1] Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Center for Research in Computing Technology. Harvard University, August 1998.

[2] Hugo Zaragoza, Djoerd Hiemstra, Michael Tipping, and Stephen Robertson. Bayesian Extension to the Language Model for Ad Hoc Information Retrieval. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4-9, 2003.

[3] ChengXiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS) 22*(2), pages 179-214, 2004.

N-GRAM MODELS

Djoerd Hiemstra
University of Twente
http://www.cs.utwente.nl/~hiemstra

DEFINITION

In language modeling, n-gram models are probabilistic models of text that use some limited amount of history, or word dependencies, where n refers to the number of words that participate in the dependence relation.

MAIN TEXT

In automatic speech recognition, n-grams are important to model some of the structural usage of natural language, i.e. the model uses word dependencies to assign a higher probability to "how are you today" than to "are how today you", although both phrases contain the exact same words. If used in information retrieval, simple unigram language models (n-gram models with $n = 1$), i.e., models that do not use term dependencies, result in good quality retrieval in many studies. The use of bigram models (n-gram models with $n = 2$) would allow the system to model direct term dependencies, and treat the occurrence of "New York" differently from separate occurrences of "New" and "York", possibly improving retrieval performance. The use of trigram models would allow the system to find direct occurrences of "New York metro", etc. The following equations contain respectively (12) a unigram model, (13) a bigram model, and (14) a trigram model:

$$P(T_1, T_2, \cdots T_n|D) = P(T_1|D)P(T_2|D) \cdots P(T_n|D) \tag{12}$$

$$P(T_1, T_2, \cdots T_n|D) = P(T_1|D)P(T_2|T_1, D) \cdots P(T_n|T_{n-1}, D) \tag{13}$$

$$P(T_1, T_2, \cdots T_n|D) = P(T_1|D)P(T_2|T_1, D)P(T_3|T_1, T_2, D) \cdots P(T_n|T_{n-2}, T_{n-1}, D) \tag{14}$$

The use of n-gram models increases the number of parameters to be estimated exponentially with n, so special care has to be taken to smooth the bigram or trigram probabilities (see PROBABILITY SMOOTHING). Several studies have shown small but significant improvements of using bigrams if smoothing parameters are properly tuned [2, 3]. Improvements of the use of n-grams and other term dependencies seem to be bigger on large data sets [1].

CROSS REFERENCE

LANGUAGE MODELS, PROBABILITY SMOOTHING

RECOMMENDED READING

[1] Donald Metzler and W. Bruce Croft. A Markov Random Field Model for Term Dependencies. In *Proceedings of the 28th ACM Conference on Research and Development in Information Retrieval (SIGIR'05)*, pages 472–479, 2005.

[2] David R.H. Miller, Tim Leek, and Richard M. Schwartz. A hidden Markov model information retrieval system. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, pages 214–221, 1999.

[3] Fei Song and W. Bruce Croft. A General Language Model for Information Retrieval. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, pages 4–9, 1999.