L

Language Models

Djoerd Hiemstra
University of Twente, Enschede, The Netherlands

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 whose 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 1980s [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 the beginning of the twentieth 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 is 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 1990s, 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 1990s [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 five papers in total [12]. In 2003, a group of leading information retrieval researchers published a research roadmap "challenges in in-

L. Liu, M. T. Özsu (eds.), *Encyclopedia of Database Systems*,
https://doi.org/10.1007/978-1-4614-8265-9

formation retrieval and language modeling" [1], indicating that the future of information retrieval and the future of language modeling cannot be seen separate from each other.

Foundations

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 term 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 $T_1, T_2,...,Tn$. Then, assuming independence between the successive events, the probability of the terms given the document D is defined as follows:

$$P\left(T_1, T_2, \ldots, T_n | D\right) = \prod_{i=1}^{n} P\left(T_i | D\right)$$

A simple language modeling approach would compute (Eq. 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 (Eq. 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* of the document model $P(T|D)$ with a collection model $P(T|C)$, which is defined as follows:

$$P\left(T_1, T_2, \ldots, T_n | D\right) = \prod_{i=1}^{n} \left(\lambda P\left(T_i | D\right) + (1-\lambda) P\left(T_i | C\right)\right)$$

In this way, a term that does not occur in the document will not be assigned a 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 Eq. 2 by two basic transformations: first, dividing it by the probability of the collection model, and second, taking the logarithm.

$$P\left(T_1, T_2, \ldots, T_n | D\right) \propto \sum_{i=1}^{n} \log\left(1 + \frac{\lambda P\left(T_i | D\right)}{(1-\lambda) P\left(T_i | C\right)}\right)$$

Equation 3 no longer produces probabilities, but it ranks the documents in the exact same order as Eq. 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:

$$P\,(D|\,T_1, T_2, \ldots, T_n) = \frac{P\,(T_1, T_2, \ldots, T_n|\,D)\,P(D)}{P\,(T_1, T_2, \ldots, T_n)}$$

The left-hand side of Eq. 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, Eq. 2 needs to be multiplied by $P(D)$ and divided by $P(T_1,...,Tn)$. Again, as stated earlier, the probabilities themselves are of no interest, but the ranking of the document by the probabilities is. And since $P(T_1,...,Tn)$ does not depend on the document, ranking the documents by the numerator of the right-hand side of Eq. 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, it might be assumed that long documents are more likely to be useful than short documents [5, 6]. 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 [10].

Document Generation Models

An implicit assumption of the language models presented 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 of tracking a stream of chronologically ordered stories. For each story in the stream, the system has to decide whether it is on topic. 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 model* approach [13]:

$$P\,(Q|\,T_1, T_2, \ldots, T_n) \propto \sum_d \left(P\,(D = d)\,P\,(Q|\,D = d) \prod_{i=1}^{n} P\,(T_1 = t_i|\,D{=}d)\right)$$

Here, the formula defines the probability of a new word Q, given the original query $T_1,...,Tn$ by marginalizing over all documents. In practice, only the top ranked documents for the query $T_1,...,Tn$ 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 an English-Dutch dictionary at random a Dutch term S (with prob-

ability $P(S|T)$). The model might be used in a cross-language retrieval system to rank English documents given a Dutch query S_1,...,$S\,n$ by the following probability [4, 6, 13, 19]:

$$P\,(S_1, S_2, \ldots, S_n|\,D)$$
$$= \prod_{i=1}^{n} \sum_{t} (P\,(S_1{=}s_i|\,T_i{=}t)\,(\lambda P\,(T_i{=}t|\,D) + (1-\lambda)\,P\,(T_i = t|\,C)))$$

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 estimated on auxiliary data. The two background models are related as follows: $P(S|C) = \sum tP(S|T=t)P(T=t|C)$. 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]: (i) select a document D with probability $P(D)$, (ii) pick a latent aspect Z with probability $P(Z|D)$, (iii) generate a term T with probability $P(T|Z)$ independent of the document, and (iv) repeat Step 2 and Step 3 until the desired number of terms is reached. This leads to Eq. 7

$$P\,(T_1, T_2, \ldots, T_n|\,D) \propto \prod_{i=1}^{n} \left(\sum_{z} (P\,(T_i|\,Z = z)\,P\,(Z = z|\,D)) \right)$$

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 probability of arriving 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 nonoptimal local maximum. As a solution, clustering of documents has been proposed to initialize the models [16]. Another alternative is latent semantic Dirichlet allocation [15] 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-References

- N-Gram Models
- Probability Smoothing

Recommended Reading

1. Allan J, Aslam J, Belkin N, Buckley C, Callan J, Croft B, Dumais S, Fuhr N, Harman D, Harper DJ, Hiemstra D, Hofmann T, Hovy E, Kraaij W, Lafferty J, Lavrenko V, Lewis D, Liddy L, Manmatha R, McCallum A, Ponte J, Prager J, Radev D, Resnik

P, Robertson S, Rosenfeld R, Roukos S, Sanderson M, Schwartz R, Singhal A, Smeaton A, Turtle H, Voorhees E, Weischedel E, Xu J, Zhai CX, editors. Challenges in information retrieval and language modeling. SIGIR Forum. 2003;37(1):31–47.
2. Balog K, Azzopardi L, Rijke M. Formal models for expert finding in enterprise corpora. In: Proceedings of 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 2006. p. 43–50.
3. Basharin GP, Langville AN, Naumov VA. The life and work of A.A. Markov. Linear Algebra Appl. 2004;386(1):3–26.
4. Berger A, Lafferty J. Information retrieval as statistical translation. In: Proceedings of 22nd ACM Conference on Research and Development in Information Retrieval; 1999. p. 222–9.
5. Blei DM, Ng AY, Jordan MI. Latent Dirichlet allocation. J Machine Learn Res. 2003;3(5):993–1022.
6. Hiemstra D, Jong F. Disambiguation strategies for cross-language information retrieval. Lecture notes in computer science. In: Proceedings of the 3rd European Conference on Research and Advanced Technology for Digital Libraries; 1999. p. 274–93.
7. Hiemstra D, Kraaij W. Twenty-one at TREC-7: ad-hoc and cross-language track. In: Proceedings of 7th Text Retrieval Conference; 1998. p. 227–38.
8. Hofmann T. Probabilistic latent semantic indexing. In: Proceedings of 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 1999. p. 50–57.
9. Jelinek F. Statistical methods for speech recognition. Cambridge, MA: MIT Press; 1997.
10. Jin H, Schwartz R, Sista S, Walls F. Topic tracking for radio, TV broadcast and newswire. In: Proceedings of DARPA Broadcast News Workshop; 1999.
11. Kraaij W, Westerveld T, Hiemstra D. The importance of prior probabilities for entry page search. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 2002. p. 27–34.
12. Kraft DH, Bruce Croft W, Harper DJ, Zobel J. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 2001.
13. Lavrenko V, Croft WB. Relevance models in information retrieval. In: Bruce Croft W, Lafferty J, editors. Language modeling for information retrieval. Kluwer: Dordecht; 2003. p. 11–56.
14. Miller DRH, Leek T, Schwartz RM. A hidden Markov model information retrieval system. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 1999. p. 214–21.
15. Ponte JM, Bruce CW. A language modeling approach to information retrieval. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval; 1998. p. 275–81.
16. Schwartz RM, Sista S, Leek T. Unsupervised topic discovery. In: Proceedings of Language Models for Information Retrieval Workshop; 2001.
17. Shannon CE. A mathematical theory of communication. Bell Syst Tech J. 1948;27(379–423):623–56.
18. Spitters M, Kraaij W. Language models for topic tracking. In: Bruce Croft W, Lafferty J, editors. Language modeling for information retrieval. Dordecht: Kluwer; 2003. p. 95–124.
19. Xu J, Weischedel R. A probabilistic approach to term translation for cross-lingual retrieval. In: Bruce Croft W, Lafferty J, editors. Language modeling for information retrieval. Dordecht: Kluwer; 2003. p. 125–40.
20. Zhai C, Lafferty J. Model-based feedback in the language modeling approach to information retrieval. In: Proceedings of ACM International Conference on Information and Knowledge Management; 2001. p. 403–10.

Languages for Web Data Extraction

Nicholas Kushmerick
VMWare, Seattle, WA, USA

Synonyms

Information extraction; Screen scraping; Web mining; Web scraping; Web site wrappers

Definition

Web data extraction is the process of automatically converting Web resources into a specific structured format. For example, if a collection of HTML web pages describes details about various companies (name, headquarters, etc) then web data extraction would involve converting this native HTML format into computer-processable data structures, such as entries in relational database tables. The purpose of web data extraction is to make web data available for subsequent manipulation or integration steps. In the previous example, the goal may be summarizing the results as some form of analytical report.