(54) Title: SEGMENTING WORDS USING SCALED PROBABILITIES

(57) Abstract: Systems, methods, and apparatuses including computer program products for segmenting words using scaled probabilities. In one implementation, a method is provided. The method includes receiving a probability of a n-gram identifying a word, determining a number of atomic units in the corresponding n-gram, identifying a scaling weight depending on the number of atomic units in the n-gram, and applying the scaling weight to the probability of the n-gram identifying a word to determine a scaled probability of the n-gram identifying a word.
SEGMENTING WORDS USING SCALED PROBABILITIES
SEGMENTING WORDS USING SCALED PROBABILITIES

BACKGROUND

[0001] This specification relates to segmenting words using scaled probabilities.

[0002] A n-gram is a sequence of n consecutive tokens, e.g., words or characters. A n-gram has an order, which is the number of tokens in the n-gram. For example, a 1-gram (or unigram) includes one token; a 2-gram (or bi-gram) includes two tokens.

[0003] Each n-gram has an associated probability estimate that is calculated as a function of n-gram relative frequency in training data. For example, a string of L tokens is represented as n-gram relative frequency in training data. For example, a string of L tokens is represented as

[0004] Each traditional word segmentation technique assumes that the probabilities of n-grams are independent. Therefore, the traditional techniques use a product of probabilities of the product of the n-gram identifying a particular word. Lesser order n-grams are derived from the n-gram. For example, suppose a n-gram is "abc". Then, lesser order n-grams of the n-gram "abc" include: "a", "b", "c", "ab", and "bc". The probability of the n-gram (e.g., "abc") identifying more than one word is the product of the individual probabilities of each lesser order n-gram identifying a word (e.g., "a", "b", and "c", "a" and "bc", or "ab" and "c").

[0005] Because the traditional techniques follow the principle of independent probabilities, the traditional techniques strongly favor segmenting n-grams into words including a greater number of atomic units than words including a greater number of atomic units. An atomic unit is a smallest ideographic unit that can be derived from a n-gram (e.g., English characters for the English language). For example, suppose a n-gram is "abc". Further assume that "a", "b", "c", and "abc" each have a probability of identifying a word equal to 0.1, or

[0006] Although the probabilities of "a", "b", and "c" each identifying a word, and the probability of "abc" identifying a word are equally likely, the traditional techniques strongly favor segmenting the n-gram into the longer word, "abc". Using traditional techniques, the probability of identifying the word "abc" is greater than the probability of identifying the individual words "a", "b", and "c".
probability of "abc" identifying three separate words (i.e., "a", "b", and "c") equals the probability of identifying three separate words (i.e., "a", "b", and "c") equals the probability of identifying a word, multiplied by the probability of identifying a word multiplied by the probability of identifying a word, or:

\[ P(\text{"abc"}) = P(\text{"a"}) \times P(\text{"b"}) \times P(\text{"c"}) \]

As a result, the traditional techniques are biased toward segmenting the n-gram into "abc". Therefore, the probability that "abc" identifies a single word is far greater than the probability that "abc" identifies three separate words. This is because the probability of identifying a word, multiplied by the probability of identifying a word, multiplied by the probability of identifying a word, is far greater than the probability of identifying three separate words. This bias towards identifying whole words can be problematic, especially when dealing with shorter words.

**SUMMARY**

Systems, methods, and apparatus including computer program products for segmenting words using scaled probabilities. In general, one aspect of the subject matter described in this specification can be embodied in methods that include the actions of receiving a word, determining a number of atomic units in the corresponding n-gram, identifying a scaling weight depending on the number of atomic units in the n-gram, and applying the scaling weight to the probability of the n-gram identifying a word.

Other embodiments of this aspect include corresponding systems, apparatus, and computer program products. These embodiments can optionally include one or more of the following features: The scaled probability of the n-gram identifying a word can depend on the number of atomic units in the n-gram. The probability of identifying a word can be scaled by the probability of identifying a word, multiplied by the number of atomic units in the n-gram.

The method can further include receiving a plurality of tokens, and segmenting the plurality of tokens into words using the scaled probability. The method can further include identifying shorter order n-grams, the shorter order n-grams being derived from the n-gram, identifying shorter order n-grams, the shorter order n-grams being derived from the n-gram.
receiving probabilities corresponding to each of the lesser order n-grams identifying words, receiving probabilities corresponding to each of the lesser order n-grams identifying words, comparing the probability of the n-gram identifying a word to the probabilities of a combination of the lesser order n-grams identifying words, and when a probability of a combination of lesser order n-grams identifying a word differs from the probability of the n-gram identifying a word by a specified threshold amount, modifying the scaling weight corresponding to the probability of the n-gram identifying a word. The method can further include receiving a scaled probability of a n-gram identifying a word, determining scaled probabilities of lesser order n-grams identifying words, the lesser order n-grams being derived from the n-gram, and removing the n-gram from a dictionary when a scaled probability of a combination of lesser order n-grams identifying a word differs from the scaled probability of combination of lesser order n-grams identifying a word by a specified threshold amount.

[0012] Particular embodiments of the subject matter described in this specification can be implemented to realize one or more of the following advantages. Segmenting words using scaled probabilities prevents problems resulting from inaccurately favoring longer words over shorter words during segmentation. Using scaling probabilities in logarithmic space to determine segmentations avoids an arbitrary application of scaling weights, which results in segmentations that do not inaccurately favor segmentations of longer words over segmentations of shorter words. The scaling weights can be pre-computed in logarithmic space, thereby improving the accuracy of segmentations without impairing runtime performance. Refined scaled probabilities can be used in segmenting, which allows n-grams to be removed from a dictionary, thereby reducing the memory size of the dictionary. Refined scaling weights can account for probabilities that are not truly independent, resulting in improved accuracy in word segmentations.

[0013] The details of one or more embodiments of the subject matter described in this specification are set forth in the accompanying drawings and the description below. Other features, aspects, and advantages of the subject matter will become apparent from the description, the drawings, and the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] FIG. 1 is a block diagram illustrating an example segmentation system for segmenting words using scaled probabilities.

[0015] FIG. 2 is a flowchart showing an example process for generating a scaled probability of a n-gram identifying a word.
FIG. 3 is a flow chart showing an example process for modifying a scaling weight corresponding to a n-gram depending on the n-gram. FIG. 4 is a flow chart showing an example process for removing a n-gram from a dictionary depending on scaled probabilities of lesser order n-grams derived from the n-gram.

FIG. 5 is a schematic diagram of a generic computer system.

Like reference numbers and designations of the various drawings indicate like elements.

DETAILED DESCRIPTION

In order to prevent word segmentations that inaccurately favor longer words over shorter words, scaled probabilities can be used to segment n-grams into words. Scaled probabilities are determined by applying scaling weights as will be described in further detail below with reference to FIG. 2. In some implementations, the probabilities of n-grams identifying words can be trained using the relative frequency of the n-grams in training data (e.g., acorpus of web pages, search log, blogs). Additionally, in some implementations, a distributed training environment is used for large training data (e.g., terabytes of data). One example technique is MapReduce. Additional details of MapReduce are described in J. Dean and S. Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, Proceedings of the 6th Symposium on Operating Systems Design and Implementation, pp. 137-150, (December 6, 2004).

In some implementations, a dictionary 100 includes a collection of n-grams 102 and corresponding probabilities. The dictionary 100 includes a collection of n-grams 102 and corresponding probabilities of the n-grams identifying words 104. Thus, for each n-gram in the dictionary, there is a corresponding probability that the n-gram identifies a particular word. The probabilities can be determined, for example, based on the relative frequency of the n-grams in the training data. Alternatively, the probabilities and n-grams of the dictionary can be received by the system as input from an external source (e.g., a language model).

Scaling weights are applied to the probabilities of the n-grams identifying words to determine scaled probabilities of n-grams identifying words. In some implementations, a scaling engine 120 uses the dictionary 100 to determine scaled probabilities of n-grams identifying words. In particular, the scaling engine 120 can apply scaling weights to the probabilities of n-grams identifying words 104 from the dictionary 100 to determine scaled probabilities of n-grams identifying words.
probabilities of n-grams identifying words 140. Determining scaled probabilities is described
below with respect to FIG. 2.

In some implementations, a segmenter 160 uses the scaled probabilities of n-grams
determining words 140 to segment a plurality of tokens (e.g., a sequence of Chinese characters)
identifying words 140 into words (e.g., Chinese words). For example, word segmenters and search engines can use
segmenter 160 to segment sequences of tokens in one or more languages e.g., Chinese,
Japanese, Korean, and Thai.

FIG. 2 is a flow chart showing an example process 200 for generating a scaled
probability of a n-gram identifying a word. For convenience, generation of scaled
probabilities of n-grams identifying words will be described with respect to a system that
determines scaled probabilities of n-grams identifying words will be described with respect to a system that
performs the generation.

The system receives 220 a probability of a n-gram identifying a word. The system
determines 240 a number of atomic units in the n-gram that corresponds to the received
probability. For example, the system receives a probability of a n-gram identifying a word,
where the corresponding n-gram includes a sequence of three Chinese characters "x/y/z". The
number of atomic units in "x/y/z" is three. As another example, the system receives a
probability of a n-gram identifying a word, where the corresponding n-gram includes a
sequence of English characters: "basketball". The number of atomic units in "basketball" is
ten.

In some implementations, atomic units correspond to grapheme clusters. A grapheme
cluster is a user-perceived character. For example, one or more Unicode characters can be
combined to make what a user considers to be a character or basic unit of a language. The
letter "C", with an acute accent "é", or "Ç", is a grapheme cluster. A user may perceive "é" as a
single character, but "C" can be represented by two Unicode code points (e.g., distinct
code points representing "C" and "É", respectively). Additional details of grapheme clusters
are described in, for example, M. Davis, Unicode Standard Annex #29: Text Boundaries,

The system identifies 260 a scaling weight depending on the number of atomic units.
The system identifies 260 a scaling weight depending on the number of atomic units.
In some implementations, the scaling weight is the number of atomic units. The system
applies 280 the scaling weight to the probability of the n-gram identifying a word to
determine 280 a scaled probability of the n-gram identifying a word. The scaling weight can be
determined depending on the accuracy of segmentations in training. For example, once
scaling weights are identified, the training data can be segmented using the scaled once
scaling weights are identified, the training data can be segmented using the scaled
probabilities, and the scaling weights can be modified if the corresponding scaled probabilities are too low for particular words.

[0029] For example, a scaled probability of a n-gram identifying a word is equal to a number of probability of the n-gram identifying a word raised to the n power, where n is the number of atomic units in the n-gram:

\[ \text{scaledProbability}(\text{word}) = \left[ \text{Probability}(\text{word}) \right]^n. \]

Rewritten using logarithmic terms, the scaled probability can be expressed as:

\[ \text{scaledLogProbability}(\text{word}) = \frac{\log \text{Probability}(\text{word})}{1 - \lambda(n - 1)}. \]

[0030] In another example, the scaling weight is a value between zero and n, where n is the number of atomic units in the n-gram. For example, a scaled probability of a n-gram identifying a word is equal to a probability of the n-gram identifying a word raised to the \(\left(\frac{1}{\lambda(n-1)}\right)\) power, where n is the number of atomic units in the n-gram and \(\lambda\) is a constant between zero (no scaling by the number of atomic units) and one (full scaling by the number of atomic units).

\[ \text{scaledProbability}(\text{word}) = \left[ \text{Probability}(\text{word}) \right]^\left(1 + \lambda(n-1)\right). \]

Rewritten using logarithmic terms, the scaled probability can be expressed as:

\[ \text{scaledLogProbability}(\text{word}) = \frac{\log \text{Probability}(\text{word})}{\lambda n}. \]

[0031] In some implementations, scaling weights are pre-computed. Furthermore, in some implementations, the scaled probabilities are determined in logarithmic space. Using a log scale simplifies computations for very small probabilities.

[0032] For example, a n-gram is "xyz". Assume "x", "y", "z", "xy", "yz", and "xyz" each have a probability of identifying a word equal to 0.1. A table below illustrates the probabilities of "x", "y", "z", "xy", "yz", and "xyz" identifying words. The table also illustrates probabilities of "x", "y", "z", "xy", "yz", and "xyz" identifying words. The table also illustrates scaled probabilities (e.g., \(\text{scaledProbability}(\text{word}) = \left[ \text{Probability}(\text{word}) \right]^\lambda\)) of "x", "y", "z", "xy", "yz", and "xyz" identifying words. In particular, the scaled probability of "xy" equals the probability of "xy" raised to the second power because "xy" has two atomic units, and the probability of "xyz" raised to the second power because "xyz" has three atomic units.

or:

\[ \text{scaledProbability}(\text{"xy"}) = \left[ \text{Probability}(\text{"xy"}) \right]^2 = 0.1^2 = 0.01. \]

[0033] Similarly, the scaled probability of "xyz" equals the probability of "xyz" raised to the third power because "xyz" has three atomic units:

\[ \text{scaledProbability}(\text{"xyz"}) \approx \left[ \text{Probability}(\text{"xyz"}) \right]^3 = 0.1^3 = 0.001. \]
The probabilities of segmentations of an n-gram do not equal the probabilities of the n-gram (e.g., Probability("xyz")=0.1 ≠ Probability("x", "y", "z")=0.001; Probability("xyz")=0.1 ≠ Probability("xy", "z")=0.01; and Probability("xyz")=0.1 ≠ Probability("x", "yz")=0.01). On the other hand, the scaled probabilities of segmentations of an n-gram equal the scaled probabilities of the n-gram (e.g., Probability("xyz")=0.001 = ProbabilityC"x", "y", "z")=0.001; Probability("xyz")=0.001 = Probability("x", "yz")=0.001; and Probability("xyz")=0.001 = Probability("xy", "z")=0.001).

Therefore, the scaled probabilities do not favor segmentations of longer words over shorter words. For example, using the scaled probabilities, a segmenter is equally likely to segment the n-gram "xyz" as "xyz", "x y z", or "xy z". On the other hand, using the un-scaled probabilities, the segmenter favors segmenting the n-gram "xyz" as "xyz" because the probability of "xyz" is greater than the probabilities of "x y z", "x y z", and "xy z", though in reality the probabilities may vary.

Probabilities of individual tokens in an n-gram identifying separate words that occur together (i.e., not being part of a single word) may not be truly independent. Therefore, the scaling weights can be further refined to account for the dependent nature of the probabilities. Scaling weights can be further refined to account for the dependent nature of the probabilities.

FIG. 3 is a flow chart showing an example process 300 for modifying a scaling weight. FIG. 3 is a flow chart showing an example process 300 for modifying a scaling weight corresponding to an n-gram. For convenience, modification of a scaling weight corresponding to an n-gram will be described with respect to a system that performs the process 300. The n-gram will be described with respect to a system that performs the process 300. The n-gram will be described with respect to a system that performs the process 300. The n-gram will be described with respect to a system that performs the process 300. The n-gram will be described with respect to a system that performs the process 300. The n-gram will be described with respect to a system that performs the process 300. The n-gram will be described with respect to a system that performs the process 300. The n-gram will be described with respect to a system that performs the process 300. The n-gram will be described with respect to a system that performs the process 300.

The system identifies 310 lesser order n-grams by deriving the lesser order n-grams from an n-gram. The system receives 320 probabilities corresponding to each of the lesser order n-grams identifying words. The system then compares 330 of the probability of the n-gram identifying a word to the probabilities of combinations 36 of the lesser order n-grams identifying a word to the probabilities of combinations of the lesser order n-grams...
identifying words. Combinations of lesser order n-grams are derived from the n-gram. For example, if an n-gram is "abc", the combinations of lesser order n-grams are "ab", "bc", "ca" (corresponding to the probability of the n-gram identifying a word by a specified threshold amount ("Yes" branch of step 340), then the system increases 350 the scaling weight corresponding to the probability of the n-gram identifying a word. Otherwise ("No" branch of step 340), the process does not modify the scaling weight corresponding to the n-gram identifying a word. For example, if a particular sequence of tokens "there of" is very unlikely, then the probability of "thereof" should be modified. In particular, the scaling weight corresponding to the probability of "thereof" identifying a word can be increased if the probability of "there of" is less than the probability of "thereof" by a specified amount.

In some implementations, the system decreases the scaling weight corresponding to the probability of the n-gram identifying a word. For example, if a particular sequence of tokens "can not" is very likely, then the probability of "cannot" should be modified. In particular, the scaling weight corresponding to the probability of "cannot" identifying a word can be decreased if the probability of "can not" is greater than the probability of "cannot" by a specified amount.

In some implementations, n-grams are removed from the dictionary under particular circumstances. The dictionary (e.g., dictionary 100) includes a collection of n-grams and corresponding probabilities of the n-grams identifying words. For example, if a n-gram in a dictionary can be segmented using two or more lesser order n-grams (derived from the n-gram in the dictionary, and a product of the scaled probabilities of the lesser order n-grams differs from a scaled probability of the n-gram by a specified threshold amount, then the n-gram is removed from the dictionary. In some implementations, if the product of the scaled probabilities of the lesser order n-grams is greater than the scaled probability of the n-gram, then the n-gram is removed from the dictionary. In the example, segmenter (e.g., segmenter 160) would not segment the n-gram as identifying a word since the probability of the particular n-gram is less than the probability of the alternative segmentation using lesser order n-grams. Removing n-grams that would not be segmented as identifying words provides a more compact dictionary.
FIG. 4 is a flow chart showing an example process 400 for removing an n-gram from a dictionary depending on scaled probabilities of lesser order n-grams derived from the n-gram. For convenience, removal of an n-gram from a dictionary depending on scaled probabilities of lesser order n-grams derived from the n-gram will be described with respect to a system that performs the process 400.

The system receives 410 a scaled probability of a n-gram identifying a word. The system determines 420 scaled probabilities of lesser order n-grams identifying words, where system determines 420 scaled probabilities of lesser order n-grams identifying words, where the lesser order n-grams are derived from the n-gram. The system compares 430 the scaled lesser order n-grams are derived from the n-gram. The system compares 430 the scaled probability of the n-gram identifying a word to the scaled probabilities of combinations of the lesser order n-grams identifying words. If a scaled probability of a lesser order n-gram identifying a word is greater than the scaled probability of the n-gram identifying a word, identifying a word is greater than the scaled probability of the n-gram identifying a word ("Yes" branch of step 440), then the system removes 450 the n-gram from a dictionary (e.g., a dictionary that includes scaled probabilities). Otherwise ("No" branch of step 440), the process does not modify 460 the dictionary, e.g., does not remove the n-gram from the dictionary.

FIG. 5 is a schematic diagram of a generic computer system 500. The system 500 can be used for practicing operations described in association with the techniques described previously, e.g., processes 200, 300, and 400). The system 500 can include a processor 510, a memory 520, a storage device 530, and input/output devices 540. Each of the components 510, 520, 530, and 540 are interconnected using a system bus 550. The processor 510 is capable of processing instructions for execution within the system 500. Such executed instructions can implement one or more components of the segmentation system of FIG. 1, for example. In one implementation, the processor 510 is a single-threaded processor. In another implementation, the processor 510 is a multi-threaded processor. The processor 510 is capable of processing instructions stored in the memory 520 or on the storage device 530 to display graphical information for a user interface on the input/output device 540.

The memory 520 is a computer readable medium such as volatile or non volatile that stores information within the system 500. The memory 520 could store the dictionary 100, for example. The storage device 530 is capable of providing persistent storage for the system 500. The storage device 530 may be a floppy disk device, a hard disk device, an optical disk device, or a tape device, or other suitable persistent storage media. The input/output device 540 provides input/output operations for the system 500. In one implementation, the device input/output device 540 includes a keyboard and/or pointing device. In another
implementation, the input/output device 540 includes a display unit for displaying graphical implementation, the input/output device 540 includes a display unit for displaying graphical user interfaces. The input/output device 540 can provide input/output operations for the segmentation system of FIG. 1. The segmentation system can include computer software components, to determine scaled probabilities. Examples of such software components include the scaling engine 120, the segmenter 160. Such software components 120 and 160 can be persistently stored in the device 550, memory 520 or can be obtained over a network connection, to name a few device 550, memory 520 or can be obtained over a network connection, to name a few structural equivalents, or combinations of one or more of them. Embodiments of the subject matter described in this specification can be implemented in one or more computer programs, i.e., one or more modules of computer program instructions encoded on a tangible program carrier for execution by, or to control the operation of, data processing apparatus. The tangible program carrier can be a propagated signal or a computer-readable medium. The propagated signal is an artificially generated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to a suitable receiver apparatus for execution by a computer. The computer-readable medium can be a machine-readable storage substrate, a memory device, a composition of matter effecting a machine-readable propagated signal, or a combination of one or more of them.

The term "data processing apparatus" encompasses all apparatus, devices, and systems. The term "data processing apparatus" encompasses all apparatus, devices, and machines for processing data, including by way of example a programmable processor, a machine for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of these.

A computer program (also known as a program, software, software application, script, of code) can be written in any form of programming language, including 'compiled' or interpreted languages, or declarative or procedural languages, and it can be deployed in any form, including as stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily run on an operating system, a combination of the or other environments.
correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, subprograms, or portions of code). A computer program can be employed to the processes and logic flows described in this specification can be performed by one or more programs or data in a file that holds a program, in a file that holds other programs or data, in one or more modules, subprograms, or portions of code. A computer program can be deployed to the computer system in which it operates, by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction.
with a user as well; for example, feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input.

[0053] Embodiments of the subject matter described in this specification can be implemented in a computing system that includes a back end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front end component, e.g., a client computer having a graphical user interface or a Web browser component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation of the subject matter described is through which a user can interact with an implementation of the subject matter described is through which a user can interact with an implementation of the subject matter described in this specification, or any combination of one or more such back end, middleware, or front end components. The components of the system can be interconnected by any form or medium of components. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication digital communication include a local area network ("LAN") and a wide area network ("WAN"), e.g., the Internet.

[0054] The computing system can include clients and servers. A client and server are generally remote from each other and typically interact through a communication network. The relationship of client and server arises by virtue of computer programs running on the respective computers and having a client-server relationship to each other.

[0055] While this specification contains many specific implementation details, these should not be construed as limitations on the scope of any invention or of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular inventions. Certain features that are described in this specification in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and/or being mutually exclusive, in certain circumstances, the subject matter described above should not be understood as requiring such separation in all embodiments.

Moreover, although features may be described above as acting in certain combinations and/or being mutually exclusive, in certain circumstances, the subject matter described above should not be understood as requiring such separation in all embodiments.

Moreover, although features may be described above as acting in certain combinations and/or being mutually exclusive, in certain circumstances, the subject matter described above should not be understood as requiring such separation in all embodiments.
and it should be understood that the described program components and systems can generally be integrated together in a single software product or packaged into multiple software products. Particular embodiments of the subject matter described in this specification have been described. Other embodiments are within the scope of the following claims. For example, the actions recited in the claims can be performed in a different order and still achieve desirable results. As one example, the processes depicted in the accompanying figures do not necessarily require the particular order shown, or sequential order, to achieve desirable results. In certain implementations, multitasking and parallel processing may be advantageous.
CLAIMS

1. A method comprising:
   receiving a probability of a n-gram identifying a word;
   determining a number of atomic units in the corresponding n-gram;
   identifying a scaling weight depending on the number of atomic units in the n-gram;
   identifying a scaling weight depending on the number of atomic units in the n-gram;
   and
   applying the scaling weight to the probability of the n-gram identifying a word to
   determine a scaled probability of the n-gram identifying a word.

2. The method of claim 1, where the scaled probability of the n-gram identifying a word
   depends on the number of atomic units in the n-gram.

3. The method of claim 1, where the scaled probability of the n-gram identifying a word
   is \( x^n \) where \( x \) is the probability of the n-gram identifying a word and \( n \) is the number of
   atomic units in the n-gram.

4. The method of claim 1, where the scaled probability of the n-gram identifying a word
   is \( x^{1+kn-1} \) where \( x \) is the probability of the n-gram identifying a word, \( n \) is the number of
   atomic units in the n-gram, and \( k \) is a constant and \( 0 < k < 1 \).

5. The method of claim 1, further comprising:
   receiving a plurality of tokens; and
   segmenting the plurality of tokens into words using the scaled probability.
6. The method of claim 1, further comprising:
   identifying lesser order n-grams, the lesser order n-grams being derived from the n-gram;
   receiving probabilities corresponding to each of the lesser order n-grams identifying words;
   receiving probabilities corresponding to each of the lesser order n-grams identifying words;
   comparing the probability of the n-gram identifying a word to the probabilities of combinations of the lesser order n-grams identifying words, and combinations of the lesser order n-grams identifying words; and
   when a probability of a combination of lesser order n-grams identifying a word differs from the probability of the n-gram identifying a word by a specified threshold amount, modifying the scaling weight corresponding to the probability of the n-gram identifying a word.

7. The method of claim 1 further comprising:
   receiving a scaled probability of a n-gram identifying a word;
   determining scaled probabilities of lesser order n-grams identifying words, the lesser order n-grams being derived from the n-gram; and
   removing the n-gram from a dictionary when a scaled probability of a combination of lesser-order n-grams identifying a word differs from the scaled probability of the n-gram identifying a word by a specified threshold amount.

8. A system comprising:
   a dictionary including n-grams and corresponding probabilities of each n-gram identifying a word;
   a scaling engine including scaling weights corresponding to each n-gram, the scaling weights depending on the number of atomic units in each n-gram; and
   scaled probabilities of each n-gram identifying a word, wherein the scaled probabilities of each n-gram identifying a word is determined including applying a scaling weight to a corresponding probability for each n-gram identifying a word.

9. The system of claim 8, where the scaled probability of the n-gram identifying a word depends on the number of atomic units in the n-gram.

10. The system of claim 8, where the scaled probability of the n-gram identifying a word is \( x^n \), where \( x \) is the probability of the n-gram identifying a word and \( n \) is the number of atomic units in the n-gram.
11. The system of claim 8, where the scaled probability of the n-gram identifying a word is
where x is the probability of the n-gram identifying a word, n is the number of
atomic units in the n-gram, and \( k \) is a constant and \( 0 \leq k \leq 1 \).

12. The system of claim 8, further comprising:

12.1. A segmenter that receives a plurality of tokens and uses the scaled probabilities
to segment the plurality of tokens into words.

13. The system of claim 8, further comprising one or more computers operable to perform
operations including:

13.1. Identifying lesser order n-grams, the lesser order n-grams being derived from the
identifying lesser order n-grams, the lesser order n-grams being derived from the
n-gram;

13.2. Receiving probabilities corresponding to each of the lesser order n-grams identifying
words;

13.3. Comparing the probability of the n-gram identifying a word to the probabilities of
combinations of the lesser order n-grams identifying words; and

13.4. When a probability of a combination of lesser order n-grams identifying a word differs
from the probability of the n-gram identifying a word by a specified threshold amount,
modifying the scaling weight corresponding to the probability of the n-gram identifying a
word.

14. The system of claim 8, further comprising one or more computers operable to perform
operations including:

14.1. Receiving a scaled probability of a n-gram identifying a word;

14.2. Determining scaled probabilities of lesser order n-grams identifying words, the lesser
order n-grams being derived from the n-gram;

14.3. Removing the n-gram from a dictionary when a scaled probability of a combination of
lesser order n-grams identifying a word differs from the scaled probability of the n-gram
identifying a word by a specified threshold amount.

identifying a word by a specified threshold amount.
15. A computer program product, tangibly stored on a computer-readable medium, comprising instructions operable to cause a programmable processor to:

- receive a probability of an n-gram identifying a word;
- determine a number of atomic units in the corresponding n-gram;
- identify a scaling weight depending on the number of atomic units in the n-gram; and
- apply the scaling weight to the probability of the n-gram identifying a word to determine a scaled probability of the n-gram identifying a word.

16. The computer program product of claim 15, where the scaled probability of the n-gram identifying a word depends on the number of atomic units in the n-gram.

17. The computer program product of claim 15, where the scaled probability of the n-gram identifying a word is \( x \), where \( x \) is the probability of the n-gram identifying a word and \( n \) is the number of atomic units in the n-gram.

18. The computer program product of claim 15, where the scaled probability of the n-gram identifying a word is \( x^{1/n} \), where \( x \) is the probability of the n-gram identifying a word, \( n \) is the number of atomic units in the n-gram, and \( a \) is a constant and \( 0 \leq k \leq 1 \).

19. The computer program product of claim 15, further comprising instructions operable to cause a programmable processor to:

- receive a plurality of tokens;
- segment the plurality of tokens into words using the scaled probability.

20. The computer program product of claim 15, further comprising instructions operable to cause a programmable processor to:

- identify lesser order n-grams, the lesser order n-grams being derived from the n-gram;
- receive probabilities corresponding to each of the lesser order n-grams identifying words;
- compare the probability of the n-gram identifying a word to the probabilities of combinations of the lesser order n-grams identifying words, and when a probability of a combination of lesser order n-grams identifying a word differs from the probability of the n-gram identifying a word by a specified threshold amount, modify the scaling weight corresponding to the probability of the n-gram identifying a word.

21. The computer program product of claim 15, further comprising instructions operable to cause a programmable processor to:

- identify lesser order n-grams, the lesser order n-grams being derived from the n-gram;
- receive probabilities corresponding to each of the lesser order n-grams identifying words;
- compare the probability of the n-gram identifying a word to the probabilities of combinations of the lesser order n-grams identifying words, and when a probability of a combination of lesser order n-grams identifying a word differs from the probability of the n-gram identifying a word by a specified threshold amount, modify the scaling weight corresponding to the probability of the n-gram identifying a word.
21. The computer program product of claim 15, further comprising instructions operable
to cause a programmable processor to:

- receive a scaled probability of a n-gram identifying a word;
- determine scaled probabilities of lesser order n-grams identifying words, the lesser
  order n-grams being derived from the n-gram, and
- remove the n-gram from a dictionary when a scaled probability of a combination of
  lesser order n-grams identifying a word differs from the scaled probability of the
  n-gram identifying a word by a specified threshold amount.

22. A system comprising:

- means for receiving a probability of a n-gram identifying a word;
- means for determining a number of atomic units in the corresponding n-gram;
- means for identifying a scaling weight depending on the number of atomic units in the
  n-gram, and
- means for applying the scaling weight to the probability of the n-gram identifying a
  word to determine a scaled probability of the n-gram identifying a word.

23. The system of claim 22, where the scaled probability of the n-gram identifying a word
depends on the number of atomic units in the n-gram.

24. The system of claim 22, where the scaled probability of the n-gram identifying a word
is \( x^n \), where \( x \) is the probability of the n-gram identifying a word and \( n \) is the number of
atomic units in the n-gram.

25. The system of claim 22, where the scaled probability of the n-gram identifying a word
is \( x^n \), where \( x \) is the probability of the n-gram identifying a word, \( n \) is the number of
atomic units in the n-gram, and \( k \) is a constant and \( 0 \leq k \leq 1 \).
FIG. 1
200°

220° Receive a probability of an n-gram identifying a word

240° Determine a number of atomic units in the n-gram

260° Identify a scaling weight depending on the number of atomic units

280° Apply the scaling weight to the probability of the n-gram identifying a word to determine a scaled probability of the n-gram identifying a word

FIG. 2
300

310 Identify lesser order n-grams from a n-gram

320 Receive probabilities corresponding to each of the lesser order n-grams identifying words

330 Compare a probability of the n-gram identifying a word to the probabilities of combinations of lesser order n-grams identifying words

340 Probabilities of the combinations of lesser order n-grams less than the probability of the n-gram identifying a word?

350 Increase the scaling weight corresponding to the n-gram

360 No modification

FIG. 3
Receive a scaled probability of a n-gram identifying a word

Determine scaled probabilities of lesser order n-grams identifying words

Compare the scaled probability of the n-gram identifying a word to the scaled probabilities of combinations of lesser order n-grams identifying words

No modification

Scaled probabilities of the combinations of lesser order n-grams greater than the scaled probability of the n-gram identifying a word?

Remove the n-gram from a dictionary