Description of the Core of the Spelling Control Software Package in the Sphinx Framework Environment

Djumanov Olimjon Israilovich
Candidate of Technical Sciences, Associate Professor, Department of Information Technologies, Samarkand State University, Samarkand, Uzbekistan

Tirkashev Farrukh Sodiq o'g'li
Graduate student, Department of Information Technologies, Samarkand State University, Samarkand, Uzbekistan

ABSTRACT

The task of constructing a software complex for the control and correction of spelling errors in texts in natural languages in the environment of the framework "SPHINX" was set. Mechanisms of structured n-grams of grammar models, identification mechanisms are developed, and descriptions of the main core of the functioning of the complex based on the XML format are given. Implemented methods for describing the main components of the n-gram grammar: grammar import, description of the lexicon, random numbers of n-grams, delay weights.

KEYWORDS: *text, document, electronic document management system, reliability, software package.*

Relevance of the topic. In various electronic document management systems, a large amount of textual information is entered, transmitted and processed. However, when sending text messages, for various reasons, errors appear that reduce the reliability of information [1]. The appearance of errors in texts manifests itself mainly in the form of spelling errors, and they can be classified as distortions of various multiplicity, for example, single (monograms), double (digrams), etc., which in the general case can be represented as n-fold [2].

There are two important directions in the construction of methods for dealing with n-fold errors. The first direction of research is based on the construction of methods for monitoring and correcting errors based on morphological analysis models [3]. However, morphological models require the involvement of a large amount of vocabulary, special word formation procedures and deep knowledge of the language being processed.

The shortcomings of existing approaches can be eliminated by building mechanisms for monitoring and correcting spelling errors based on little-studied n-grams of language models [4,5]. Moreover, the use of n-grams provides more flexible solutions in various applications such as information retrieval, machine translation, etc.

In this paper, we propose a solution to the problem of increasing the reliability of the transmission and processing of texts based on the creation of mechanisms for representing n-grams, a stochastic grammar format for maintaining a dictionary and open dictionary sentences, methods for describing a lexicon and grammar classes.

The representation of the grammar of a language based on n-grams is reduced to the mapping of the nth order of the Markov model of the language, where the probability of the appearance of the

desired character is due to the appearance of other characters preceding the $(n-1)$.

An important feature of n-gram models is that they allow you to cover a wider range of words of the language than recorded directly in the corpus under study, which is typical for spelling control systems based on morphological analysis models [6].

The primary task in building a spelling control system based on n-gram models is the recognition of text elements (letters, symbols, words). It is of interest to describe the main core of the functioning of model n-grams within the system under consideration, including the identification of procedures for using model n-grams in recognition devices; combining special n-gram rules at different levels of recognition; recognition of the structure of phrases in the form of numerical expressions [7,8].

Tree-like representation model of grammar n-grams. In the representation of grammar n-grams, we propose a more flexible way of presenting basic statistical information with a random number of word sequences. The proposed method allows reducing the file size by eliminating some redundancy in the structure of its format, organizing a convenient order for data processing and loading flows, and manipulating the tag tree of the main core. In this regard, manipulation procedures are proposed to reduce redundancy based on the grammar tree [9].

The file format contains the lines of a tuple of data representing each branch and previous node of the grammar tree. The branch data is a list of indices representing the sequence of n-gram words. The subsequent word sequence data is a list of one or two integers representing the branching node index and a random number. The first entry of each format string is 'zerogram', a virtual empty branch with the successor node being the actual root of the grammar tree. From the root node, the data of the monogram node comes out, then the data is transmitted to the branches and nodes for the $p = 2,3,...,L$ levels, until leaf L, the required required depth [10].

Let us introduce a general condition for reducing the redundancy of $L < n$, and a change in this condition, in turn, leads to a change in the order of the n-grams of the model. To further reduce redundancy, a similar manipulation is carried out with passing data for members of the set to depth $(L-1)$, $(L-2)$, and so on.

Construction of a format for describing N-grams of a grammar. *Due to the fact that we have adopted the construction of a system for monitoring and correcting spelling in the SPHINX environment, it is advisable to describe n-grams of models based on the XML grammar used to describe new created procedure formats [11].*

Following XML conventions, the language and variant are denoted by the "xml:lang" attribute, which is the root of the "grammar" element tree. The developed scheme provides for importing this description into the parent n-gram or importing, in turn, into other description n-grams according to certain import rules. The lexicon that is required if the number of n-grams is defined contains the definition of an index of characters that can represent events, i.e. words or references to other grammars and grammar rules.

Random n-gram numbers are presented in first-to-depth order format. Precomputed holding weights and distant or regular n-grams are announced.

Grammar import. *The description of the import procedure is used to transfer the lower order grammar components to the senior order grammar. The import of the lower n-gram grammar declares to add additional random numbers to the set of random numbers of n-grams in senior grammar. If necessary, the entire higher grammar of n-grams can be built exclusively from imported grammars.*

This is especially useful for applying a variable number of reduction calculations to a server using

CGI requests, i.e. thus, the default abbreviations are changed, which saves unloading time.

Importing the delay weights is not desirable, since modifying the random numbers completely changes the entire set of delay weights, which must be recalculated after all the n-gram random numbers have been compiled.

An arbitrary number of import rules can be arbitrarily declared as follows:

<import uri="protocol://host/path/path_info?query_string"

name="namestring"/>

The *name* attribute is optional. The absence of an attribute name indicates that the imported grammar, which must be an n-gram grammar, will be treated as a contribution to the higher n-gram grammar and added to the higher n-gram grammar's random number list. The presence of the *name* attribute indicates that the imported grammar is the inferior grammar that the superior grammar will refer to as described later.

If the import description is set to:

<import uri=

"http://www.example.com/ngram.pl/mygrammar.g?depth=3"/>,

then the Perl script ngram.pl will need to be able to parse any line from an n-gram random number file, such as mygrammar.g, to produce a proper XML formatted description of the n-grams during parsing of the file's contents at the server.

If the import description is given as:

<import uri=

"http://www.grammars.com/cities-states.xml"

name="places"/>

... <gramref import="mygrammar"/> ...

... <ruleref import="places#start"/> ...

then this is a simple transfer of the file of the lower named grammar of n-grams to the older one. Note that for grammar n-grams, several start symbols are predefined and they are described in the corresponding < ruleref ... > tag.

Description of the lexicon. *The description of the n-gram lexicon consists of a single set of tags containing lexical entries and is necessary to determine the indices of successful n-gram random number rules. A lexical entry may contain a word character or a link rule. Link rules always refer to an external lower grammar rule [12-14].*

A dictionary can optionally be declared in two ways. The first method is based on heuristic character indexing. Characters must be indexed with non-negative integers. The numbering should be contiguous to minimize the storage required for indexing, but this is not required and the data can be presented in any specific order. Another way is to have the order="sequential" attribute, which makes the indexing sequential, starting from 1.

In both proposed methods, we have shown four examples of declaring word symbols.

Description of random numbers of n-grams. Practice shows that the n-gram grammar is rather cumbersome and requires very large file sizes when described in XML formats, which is burdensome for communication systems. In this regard, for an efficient and compact representation of grammar n-

grams, we propose a modification of the XML format representation for describing random numbers of n-grams.

The nullgram information represents the root node of the tree. In this case, 5 character instances of three different character case types are possible. Leaf nodes are indicated by the definition of zero lower branches. Since this value is always zero in leaves, this information can simply be removed. Character types can be superword characters or even grammar examples. For example, if "A B " is a character, then in the pseudo-corpus there are 3 character instances for two character types. In the case when such characters are selected, for example "A A", then only the strings "A" are possible in the corpus.

Internal separators - commas and semicolons are used to indicate the end of the n-gram rule. Whitespace is not essential within opportunities the scope of the <tree>. Note that if the reduction has been performed, then the branch values must be recomputed. The depth of the tree is implied by the data structure.

The <tree> element may have an additional 'gap' attribute, which is used for distant n-grams. Distant or jumping n-grams are used to span the dependency with n-gram models with small n. For this, a gap of some length is used between the word and its prehistory.

For the "A B C D E F G H" corpus, the regular trigram pattern provides a count of the events "A B C", "B C D", and so on. The distant n-grams are stored in the same tree structure as the regular n-grams. If the default value of the optional parameter 'gap' is zero, then this is identical to the regular n-gram model.

Description of the holding scales. *The declaration of retaining weights is necessary to distinguish between distant and regular n-grams in the description of the lexicon. Weight separators are colons. Delaying weights are applied only to non-sheet elements. Weights are calculated in ARPA format. In addition to the floating point format, a scalable integer format is supported. This changes the <tree> element to include the scale attribute.*

Model interpolation. To simulate an n-gram model, the total probability P(w|h) of the word w occurring after the prehistory h is calculated.

In this case, the n-gram can be built based on a linear interpolation of several models. In this case, the arithmetic average of P(w|h) is calculated for each of the models, and if λ_i is the normalized weight of each model, then

$P_{lin\_int}(w \mid h) = \sum \lambda_i P_i(w \mid h)$, where $\sum \lambda_i = 1$.

For identification, interpolated models are represented by the <interpolation> element.

It contains multiple <component> elements that represent each model. The 'weight' attribute on the <component> element is used to determine the relative weight of each model. The sum of all weights for each <interpolation> element must not exceed 1.0, and the platform is responsible for normalization.

Each of the <component> models defines its own lexicon, and the platform merges these lexicons automatically.

To identify the n-gram, it is possible to use the log-linear interpolation model

$P_{\log\_int}(w \mid h) = \prod P_i(w \mid h)^{\lambda_i} / Z(h)$,

where $Z(h)$ is a sign of normalization. In this case, the 'type' attribute on the <interpolation> must be

set to "log".

Conclusion. The developed scientific and methodological foundations for identifying the main core of the functioning of the n-gram model in the SPHINX environment are the key points in building the interface of the software complex for monitoring and correcting spelling in texts. It is recommended to carry out a description of the main links of the kernel based on a tree-like representation model of the n-gram grammar.

Techniques for describing the main components of an n-gram grammar have been developed: grammar import, description of the lexicon, random numbers of n-grams, and delay weights. Identification methods based on interpolation of several n-gram models are proposed.

References

1. Морозкина, Е. А. Формализация естественного языка в машинном переводе с опорой на дополнительный корпус родственного языка.
2. Карпович, С. Н. Математическое и программное обеспечение вероятностного тематического моделирования потока текстовых документов.
3. Камилов, М. М. (2010). Система контроля достоверности текстовой информации на основе n-граммных парсинговых моделей. Проблемы информатики, (1), 42-51.
4. Jumanov, I. I., Djumanov, O. I., & Safarov, R. A. (2021, September). Methodology of Optimization of Identification of the Contour and Brightness-Color Picture of Images of Micro-Objects. In 2021 International Russian Automation Conference (RusAutoCon) (pp. 190-195). IEEE.
5. Isroil, J., & Khusan, K. (2020, November). Increasing the Reliability of Full Text Documents Based on the Use of Mechanisms for Extraction of Statistical and Semantic Links of Elements. In 2020 International Conference on Information Science and Communications Technologies (ICISCT) (pp. 1-5). IEEE.
6. Jumanov, I. I., & Karshiev, K. B. (2020, May). Mechanisms for optimization of detection and correction of text errors based on combining multilevel morphological analysis with n-gram models. In Journal of Physics: Conference Series (Vol. 1546, No. 1, p. 012082). IOP Publishing.
7. Akhatov, A. R., & Jumanov, I. I. (2006, September). Improvement of text information processing quality in documents processing systems. In 2006 2nd IEEE/IFIP International Conference in Central Asia on Internet (pp. 1-5). IEEE.
8. Jumanov, I. I., Safarov, R. A., & Xurramov, L. Y. (2021, November). Optimization of micro-object identification based on detection and correction of distorted image points. In AIP Conference Proceedings (Vol. 2402, No. 1, p. 070041). AIP Publishing LLC.
9. Жуманов, И. И., & Шарипова, М. (2013). Оптимизация контроля орфографии узбекского языка основе моделей стохастического поиска. Современные материалы, техника и технология (pp. 129-133).
10. Djumanov, O. I., Kholmonov, S. M., & Shukurov, L. E. (2021). Optimization of the credibility of information processing based on hyper semantic document search. Theoretical & Applied Science, (4), 161-164.
11. Холмонов, С. М., & Абсаломова, Г. Б. (2020). Методы и алгоритмы повышения достоверности текстовой информации электронных документов. Science and world, 43.

12. Холмонов, С. М., & Абсаломова, Г. Б. (2020). Повышение достоверности текстов на основе логических критериев и базы знаний электронных документов. Технические науки: проблемы и решения (pp. 15-19).
13. Жуманов, И. И., & Каршиев, Х. Б. (2019). Оптимизация достоверности информации на основе базы электронных документов и особенностей правил контроля базы знаний. Проблемы вычислительной и прикладной математики, (3 (21)), 57-74.
14. Jumanov, I. I., Djumanov, O. I., & Safarov, R. A. (2021, November). Mechanisms for optimizing the error control of micro-object images based on hybrid neural network models. In AIP Conference Proceedings (Vol. 2402, No. 1, p. 030018). AIP Publishing LLC.