

Sentence analysis using a concept lattice

Abstract

Grammatically incorrect sentences result either from an unknown (possibly misspelled) word, an incorrect word order or even an omitted/redundant word. Sentences with these errors are a bottle-neck to NLP systems because they cannot be parsed correctly. Human beings are able to overcome this problem (either occurring in spoken or written language) since they are capable of doing a semantic similarity search to find out if a similar utterance has been heard before or a syntactic similarity search for a stored utterance that shares structural similarities with the input. If the syntactic and semantic analysis of the rest of the input can be done correctly, then a 'gap' that exists in the utterance, can be uniquely identified. In this paper, a system named SAUCOLA which is based on a concept lattice, that mimics human skills in resolving knowledge gaps that exist in written language is presented. The preliminary results show that correct stored sentences can be retrieved based on the words contained in the incorrect input sentence.

1.0 Introduction

Grammatically incorrect sentences result from either an unknown (possibly misspelled) word, an incorrect word order or even an omitted/redundant word. Humans are, nevertheless, able to infer the correct meaning of a sentence and other related information and hence facilitate continued flow of communication. This is made easy by the valuable sentence database that humans accumulate over time. The sentence database is augmented by the knowledge of the language which implies the knowledge of syntax (i.e. how words and their types follow each other to form valid sentences) and semantics (i.e. vocabulary of words and their meanings). Thus, syntax, semantics and sentence database are used in interpreting any ill-formed input sentence that can be perceived to deduce the correct meaning.

Relying on a syntactic and semantic analysis to match the input against stored knowledge, a 'gap' that may exist may often be uniquely identified. Adequate vocabulary is essential to enable a missing word to be correctly guessed. And knowledge of syntax, on the other end, makes it possible for a type of the missing word to be uniquely identified. Much knowledge and lots of example sentences are necessary since 'the more comparable objects which are known, the more a query object can be related to database objects, and hence direct the search towards the location in the database where the object fits' (Ellis 1993).

The human mind may be perceived as a complex network that processes natural language. This language network connects abstractions of words, their meanings, syntax of the language and even dynamic interconnections of words to phrases and sentences that have been conceived before. For instance, humans remember idioms, quotations and what others have said in a conversation. They are able to link words to situations or events that have happened in the past. A name, for instance, might trigger a memory of someone familiar. We are able to distinguish a new word or a phrase, because it is not in the network. When an incorrect sentence is heard/read, something 'deep inside' tells one that either a verb or a noun is missing from the heard/read sentence. We are able to do this because we can

always remember a sentence that shares syntactic similarities with an incorrect one that we are hearing/reading. When processing natural language, humans refer to this network (perhaps subconsciously) for utterance processing.

This paper presents a Sentence Analysis Using a Concept Lattice (SAUCOLA) system, based on a concept lattice (CL), for processing grammatically incorrect natural language sentences. It can be a useful augmentation to machine translation (MT) systems since part of the translation involves re-arranging of words so that they form comprehensible sentences in the target language. In conventional rule-based MT systems, word re-arrangement often occurs as a result of successively applying a series of rule re-writing or refinement. Constructing and maintaining such rule base significantly increases complexity of such systems. The SAUCOLA system aims to reduce this complexity by assuming responsibility for word re-arrangement, based on the good sentences retrieved from the stored knowledge.

Sentences also fail to parse due to an error in the input and handling of grammatically incorrect input is a concern to MT researchers (cf. (Knight et al. 1995), (Daelemans 1993), (Yamada et al. 1995)). The ability to auto correct sentences by putting words that may be incorrectly arranged into an arrangement that will be recognised by an MT system's parser for correct parsing and subsequent translation is, therefore, of prime importance.

In this paper, a CL is used to link words and their types to sentences (knowledge base) in which they have been used. The meets and the joins of those words in a CL help in locating a sentence in the knowledge base that is as close to the input as possible (nearest or closest neighbour). This similar sentence will be taken as the best approximation of the input. This approach is an attempt to model human network for processing natural language so that grammatically incorrect sentences can be fairly handled. A concept lattice, through its embedded indexing mechanism, is able to give nearest neighbours of the input sentence. When a nearest neighbour has been selected, the input can then be transformed accordingly into a valid and sensible sentence.

The rest of the paper is structured as follows: section 2 advocates the appropriateness of a CL. Section 3 deals with issues in sentence analysis of applying concept lattices to finding candidate sentences and how they can be applied in handling the under and over generations in MT. These will be demonstrated for the Sesotho¹ language. Finally, section 4 gives a brief summary of the work done and in section 5, a statement about future work is presented.

2.0 Why a concept lattice?

Much research effort in machine learning (ML) has gone into classification. Though the aims were not to support NLP applications, the resulting algorithms are useful. The classification algorithms can also be seen in a broad sense of partial matching since they can only predict the class an entity is believed to fall in. Selecting a sentence that is as close as possible to the input is also a classification task, and thus, NLP research can benefit out of the developed algorithms. This shift is based on the premise that a parse tree can be regarded as a series of classification problems.

¹Sesotho is an African language spoken in Lesotho and many other parts of the Republic of South Africa.

Demiroz and Guvenir (1997) relied on probability to predict classes into which input sentences belonged. For each class, a probability was calculated and the class that scored the highest probability was taken as the best approximation of the input sentence. In Cardie (1996) and Daelemans et al. (1997), ambiguity resolution is treated as a classification problem. This approach resulted in a range of subproblems in sentence analysis that used to be handled independently to be addressed all at once.

Classification techniques using a CL have mainly been applied in information retrieval and extraction. CL's are attractive for classification tasks because they are able to represent all inter-dependencies between features to be recorded and hence a complete inventory among features is obtainable (Oosthuizen 1988). Also, the CL is able to contain an exhaustive set of all clusters generated from objects with the named features. Therefore, given a set of features, a set of objects is directly accessed. This enables an entity that is as close as possible to the input to be selected. An input sentence must have as many words/word types as possible in common with its candidate. That is, the one with the largest number of words in common with the input will be selected as the closest neighbour.

It is important that NLP systems are capable of tackling the problem of missing 'definitive knowledge or knowledge gaps'. To fill a knowledge gap, KBMT systems make a random decision (Knight et al. 1995). These random decisions can be guided by mechanisms based upon prediction and partial matching.

3.0 Sentence analysis

Conventional MT systems are generally guided by statistics in order to choose appropriate word senses. In some instances, this is inaccurate. That is, the sense associated with the highest statistic does not necessarily mean that it is the correct one. The same is true for MT systems that select the sense of a first word in a dictionary, which is also not necessarily correct. Picking a word with the highest statistic or because it appears before others in a dictionary can therefore be a bottleneck since an incorrect word type choice would lead to a non-existent sentence construct in a language and hence failure for such a sentence to parse.

3.1 Concept lattices in sentence analysis

The use of a CL can provide a different way to handle ambiguity. During parsing, a word is neither selected according to its position in the dictionary (first word first) nor according to some rule (e.g. most commonly used term), but according to the expected type at that position in a sentence. To be able to do this, we need to take into account the words (and their types) that are already known in the sentence. Humans interpret ambiguous sentences taking into account what is already known in that sentence.

A lattice structure is independent of the order in which objects are added and therefore the words are not obtained on the first come first serve basis. The first word sense found does not necessarily serve as the correct one, but the semantic role is selected in association with other words/types in the sentence. This is done by exploiting the learnt word order, in the form of word type vectors. These are vectors made up of a linear collection of word types corresponding to a given sentence. For instance, a WTV for 'she, looked, around, the, corner, to, see, a, dog' is '<preposition, verb, adverb, article, noun, infinitive, verb, article, noun>'.

WTV's are analogous to concepts (in ML sense) since each WTV is represented by a tuple $\{S_i, W_i\}$, where S_i (an extent) is one of possible language sentences and W_i (an intent) is a linear collection of word types used to construct S_i . Since a sentence is a linear collection of identified words in a particular order, it is by definition, a chain.

Graphically, a concept lattice can be viewed as an acyclic graph linking attributes (words) to their entities (sentences). The so-called GRAND algorithm, developed by (Oosthuizen 1988) provides a way of setting up this 'sentence lattice'. The resulting sentence lattice can be viewed as a way of abstracting and organising S so that natural groupings of words emerge as internal concept nodes in the lattice. Because of the lattice properties, any pair of nodes has a unique meet and a unique join. In particular, entities that have words in common will have a unique meet. Such a meet is also, in turn, the join of all attributes corresponding to the words that the entities have in common.

In this scenario, all entities that have those words in common have a path to this meet. Naturally, a mapping from input set G onto sentence database, S , is suggested. A CL enables this mapping, whose result is a set of candidate sentences C (which is obtained by taking the meets of words in G), to be defined.

Let this mapping be:

$$\chi^s: G \rightarrow P[S]$$

where: S is a sentence database.

$P[S]$ is the power set of S .

G is a set (possibly singleton set) of input sentence

χ^s is a function that returns a result based on downward closures.

When applied to an argument $g \in G$, the function, χ^s , returns a set, C , of candidate sentences, in the sentence lattice derived from S . C is found by determining the meet of all attributes (i.e. words) in g , and then determining the set of sentences in the downward closure of this meet.

3.2 Selecting the Best Neighbour

Serutla and Oosthuizen (1997) suggest that input that cannot be parsed correctly by an MT system normally has one of the following errors: unknown words(s), redundant word(s), omitted word(s), or even incorrect word order. These errors make it impossible for an otherwise applicable rule to match and hence ensure that the input cannot be parsed. Similar errors also occur in the generated or translated sentences of a rule-based MT system. Next, the notion of rank is used to illustrate how the SAUCOLA system could be helpful in resolving under-generation (omitted words), over-generations (redundant words) and incorrect word order in sentences generated by an MT system.

Let

$g \in G$ be some generated sentence.

$\chi^s(g) = C$ is the set of candidate sentences available in S .

$c \in C$ is an arbitrary sentence in C .

$r(s)$ denote the rank of an arbitrary sentence s . This is defined as the total number of words in s .

Then the following cases may arise:

1. $r(g) < r(c)$ for all $c \in C$, i.e. the input/generated sentence is shorter than its closest neighbour. This case arises when an input/generated sentence contains at least one missing word. In the context of the present discussion, the translation mechanism failed to generate all the appropriate words required to make the translation sound. This is referred to as under generation, since not all the necessary words have been generated by the translation mechanism. For instance, the Lexica² translation of the input

"while the daughter should be trying to read in the garden" (1)

resulted in

"moralilokela be leka ho balile ka teng the ts'imo" (2)

while the correct translation should be

"ha morali a lokela hore e be o ntse a leka ho bala ka serapeng" (3)

It has been observed that if a sentence in (2) is given as an input to the SAUCOLA system, then the sentence in (3) is retrieved. It is therefore possible to retrieve correct sentences given incorrect input ones only if the sentence database contains many semantically and syntactically correct sentences.

2. $r(g) > r(c)$ for all $c \in C$, i.e. the input/generated sentence is longer than its closest neighbour. This case arises when an input/generated sentence has at least one redundant word. And this case is referred to as over generation, since more words than necessary have been generated with the consequent that correctness of the translation is clouded. To remedy this situation, words that are not required need to be pruned off leaving only those that are necessary for plausible translation. This problem also occurs in the form of repeated words and/or phrases in the input to MT systems.

The retrieved best approximation of a sentence will help in deciding which word types are not necessary in order that the input can be transformed into an acceptable form. It should be noted that the sentence database contains only those sentences that are 'syntactically and semantically correct'.

An example:

input: but the horse may try to always be fed by the boy first.

output: empa pere e ka 'ka' nna ea leka ho 'be ho' feptjoa ke moshanyana pele kamehla.

required: empa pere e ka nna ea leka ho feptjoa ke moshanyana pele kamehla.

The quoted words in the output are the over generated ones. If they are removed, the resulting sentence becomes acceptable. Again the meet of most of the words in the output point to the required sentence.

²Lexica is a transfer rule-based MT engine developed at the university of Pretoria. It is used to translate illustrative Sesotho sentences used in this paper.

3. $r(g) = r(c)$ for some $c \in C$. Here the input sentence has the same length as its closest neighbours. g and c are likely to share the same, or nearly the same, syntactic structure. However, one or more of the words in the translated sentence might have been incorrectly selected - either from a dictionary on a first come first serve basis, or from a list of alternatives on the basis of statistical evidence.

But this is not a correct interpretation of how language behaves. It is true that some words are used more frequently than others, and tagging them like that has worked for MT. The reality, however, is that selection of words is not done in isolation, but is based on what other words are already used in such a sentence. The illustration of this is a choice of a concord when translating Sesotho into English, as can be seen in the following example:

input: I am singing and chewing.
output: Nna kea bina le kea hlafuna.
required: Nna kea bina hape kea hlafuna.

In the above example, the concord 'le' (one of translations of 'and') is the first in the dictionary and hence gets picked up and used without paying any attention to the meaning, or what other words in a sentence dictate. But with the current system, the required (as in above) is retrieved given the output (also as in above).

4.0 Conclusion

This paper is based on the notion that sentence interpretation is a search problem, the search space being the set of all possible sentence interpretations. It is noted that an input sentence can be ill formed with the result that intended meaning is distorted. A mechanism has been proposed to search for the nearest sentence to the input. If found it can be used as a basis to auto-correct the input sentence.

Humans are able to auto-correct such sentences because they have accumulated a valuable sentence database over time. This is coupled with their knowledge of the language syntax and semantics. All these tools are used in interpreting an ill-formed input sentence, so that its correct meaning can be perceived or deduced.

These preliminary results show that even before considering the semantic roles that the words play in various sentences, the retrieved closest neighbours are really similar in many respects to the input sentences. We are taking the structure of the (possibly ill-formed) input sentence and looking for a corresponding structure in our language network. If this structure has been perceived before, it is retrieved; otherwise we perform a network or lattice search for a similar structure in order to interpret the currently unknown input. Eventually, we will bring both the syntax (in the form of WTV's) and the semantics (words and their types) to help us uniquely determine the unknowns in the input and hence interpret a sentence.

5.0 Future Directions

Since its introduction by Nagao(1984), example-based machine translation (EBMT) has been an active research area in NLP and some research efforts have been published (e.g. Pangloss (Nirenburg 1995), ReVerb (Collins and Cunningham 1996), ALT-J/E (Kaneda et al. 1996)). The current approach appears suited in an EBMT context. A source sentence database would have to be aligned with corresponding sentence translations in the target language.

The advantage of this is two fold: firstly, if a source sentence has been perceived before, the aligned translation becomes the required target. It need not be generated. Secondly, if the input sentence shares syntactic similarity (similar WTV) with a stored sentence, then the knowledge used to transform the original source into the aligned target can be used to transform the current input into the required target.

5.0 References

1. Cardie, C.: Embedded Machine learning systems for natural language processing: A general framework; In Wermter, S., Riloff, E., Scheler, G.(eds.). LNAI 1040. Springer(1996) 315 - 328
2. Collin, B., Cunningham, P.: Adaptation-guided retrieval in EBMT: A case based approach to machine translation. Proceedings of EWCBR'96. LNAI 1168. Springer (1996). 91 - 104
3. Daelemans, W.: Memory-based lexical acquisition and processing. Proceedings of EAMT Workshop. LNAI 898. Springer (1995) 85 - 98
4. Daelemans, W., van den Bosch, A., Weijters, T.: Empirical learning of natural language processing tasks. Proceedings of ECML'97. Springer (1997) 337 - 344
5. Demiroz, G., Guvenir, H. A.: Classification by voting feature intervals. Proceedings of ECML'97. Springer (1997) 85 - 92
6. Ellis, G.: Efficient retrieval from hierarchies of objects using lattice operations. Proceedings of ICCS'93. Springer-Verlag (1993) 274 - 293
7. Kaneda, S., Almuallim, H., Akiba, Y., Ishii, M., Kawaoka, T.: A revision learner to acquire selection rules from a human made rules and examples. LNAI 1040. Springer (1996) 439 - 452
8. Knight, K., Chander, I., Haines, M., Hatzivassiloglou, V., Hovy, E., Iida, M., Luk, S. K., Wjiteney, R., Yamada, K.: Filling knowledge gaps in a broad based coverage machine translation system. Proceedings of IJCAI'95. 1390-1396
9. Nagao, M.: A framework for mechanical translations between Japanese and English by analogy principle. In elithorn, A. and Manerji, R. (eds.): Artificial and Human Intelligence. Elsevier Publishers (1984).
10. Nirenburg, S.: (ed.) The PANGLOSS Mark III machine translation system. CMU-CMT-95-145 (1995) Joint Technical Report
11. Oosthuizen, G.D.: The use of a lattice in knowledge processing. Ph.D. Thesis (1988). University of Strathclyde. Glasgow
12. Serutla, L., Oosthuizen, G.D.: Using a lattice to enhance adaptation-guided retrieval in example based machine translation. Proceedings of SAICSIT'97. 177 - 191
13. Yamada, S., Nakaiwa, H., Ogura, K., Ikehara, S.: A method of automatically adapting MT system to a different domains. Proceedings of TMI'95 (1995) 303 - 310