DOCUMENT RESUME

ED 380 996 FL 022 819

AUTHOR Aretoulaki, Maria; Tsujii, Jun-ichi
TITLE An ANN That Applies Pragmatic Decision on Texts.
PUB DATE 9 Dec 94
NOTE 7p.
PUB TYPE Reports - Descriptive (141)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Abstracting; Abstracts; Artificial Intelligence; Classification; *Computational Linguistics; Computer Software; *Discourse Analysis; Foreign Countries; Language Patterns; Language Research; Linguistic Theory; *Pragmatics; Program Descriptions; *Sentence Structure
IDENTIFIERS *Neural Networks

ABSTRACT

A computer-based artificial neural network (ANN) that learns to classify sentences in a text as important or unimportant is described. The program is designed to select the sentences that are important enough to be included in composition of an abstract of the text. The ANN is embedded in a conventional symbolic environment consisting of lexical/semantic, morphological, syntactic, and pragmatic analyzers and synthesizers. Only certain features are computed by these symbolic modules, those that are more relevant to this sentence classification task. The selected features are translated into vectors of ones, zeros, and intermediate values, and input into the network. In this way, the ANN collectively considers all linguistic and pragmatic levels in making a decision. This is in contrast to most text abstraction systems, which use information retrieval techniques such as keyword extraction. Preliminary testing shows the ANN has a success rate of 86 percent with 10 different sets of novel sentences, after having been trained on disparate sets of 90 sentences. Contains 12 references. (Author/MSE)

An ANN that Applies Pragmatic Decisions on Texts

Maria ARETOULAKI * and Jun-ichi TSUJII

Centre for Computational Linguistics,
Dept. of Language and Linguistics,
UMIST,
Sackville Street, PO Box 88,
Manchester M60 1QD, U.K.
Tel: (61) 200 3100
Fax: (61) 200 3099
Email: [mariaa,tsujii]@ccl.umist.ac.uk

Abstract

A feedforward/BP ANN is described that assigns degrees of importance to the sentences in a text in order to select the ones that will be used in the composition of its abstract. The higher the importance rate the more likely it is for the corresponding sentence to be included in the abstract. The ANN is embedded in a conventional symbolic environment consisting of lexical/semantic, morphological, syntactic and pragmatic analysers and synthesisers. Only certain features are computed by these symbolic modules, those that are more relevant to this sentence classification task. The selected feature values are translated into vectors of 1s, 0s and intermediate values, and input to the network. By consequence, the ANN collectively considers all linguistic and pragmatic levels in taking a decision. This is in contrast to most text abstraction systems to date, which employ keyword extraction and other such information retrieval techniques. Preliminary results show that the ANN has an 86% success rate with 10 different sets of novel sentences, after having been trained on disparate sets of 90 sentences.

INTRODUCTION

The primary goal of this research is the identification of those linguistic and extra-linguistic (i.e. pragmatic) features that determine *importance* in a sentence. The emphasis is on *universality* and *domain-independence*. The features should be applicable to unrestricted, real-world texts. In the long run, the sentences selected as more important will be the ones that will be used in the construction of the *abstract* of the corresponding text. Two major assumptions are involved:

- Importance does *not* depend on individual features. Instead, *feature patterns* need to be discovered that are both necessary and sufficient in inducing importance in a sentence. Analysis of real-world text corpora will eventually single out the most salient features and feature combinations.
- To date, there are *no strong* or *clear-cut rules* for this sentence classification task. Furthermore, it is unlikely that any such rules can be readily established, due to the strong dependence of each sentence on its individual context. To overcome this hindrance, an ANN was introduced which processes features the values of which take into account previous phrases and sentences.

Text Abstraction

Text Abstraction constitutes the reduction of a text to only those propositions that convey the *most crucial messages* of the text. It is not just a reduction of the physical presence of the text, but also of its content. This is potentially a very useful application, because it could provide a solution to the current problem of *information overload*. Abstracts could help researchers, for example, *filter* numerous pages of electronic text from various academic newsgroups, bulletin boards and journals, and choose the messages or articles that need to be read in full. Thus they can eliminate the time and effort spent on the irrelevant and put additional time on the necessary and the interesting.

The process of abstraction involves two operations:

1. The *selection* of the *important* sentences or propositions. These provide the subject matter for the abstract itself.
2. Their *assimilation* into a short, but coherent and cohesive text, the final product of abstraction.

*This research has been supported by a CANON Europe PhD studentship.

The current focus of this research is on the first task, that of choosing the important sentences and, by definition, also identifying the feature patterns that entail importance.

Most abstraction systems to date employ standard information retrieval techniques, such as *keyword matching* and *lexical pattern* matching in deciding which sentences to consider for the final summary (Mauldin et al., 1987). These methods can be very effective but only as regards texts of a well-defined structure and limited syntax and vocabulary, e.g. telex messages (Young & Hayes, 1985). When more complex texts are involved, such as newspaper articles, success is limited, because neither the vocabulary nor the syntax can be fully predicted and accounted for (Appelt et al., 1993). On the other hand, more sophisticated research invariably focuses on theoretical, discourse analytical, issues that rarely lead to a working system (Lucas et al., 1993). In contrast, a *holistic* approach has been adopted here, which dictates the *cummulative* consideration of all, and not just the lexical and syntactic or the pragmatic, levels in selecting the important sentences in a text.

AN ANN FOR PRAGMATICS

In order to discover the feature patterns pertinent to sentence importance, a *feedforward/BP* Artificial Neural Network *(ANN)* (Rumelhart et al., 1986) is employed that learns to classify sentences into important and unimportant. This is done on the basis of pairs of input feature value vectors and output importance ratings that have been taught to the network in a *supervised* manner. The input features have been computed by symbolic linguistic and extra-linguistic analysers, which attribute certain lexical-semantic, morphological, syntactic and pragmatic values to each sentence in a text. The output vector consists of just two values, one for importance and one for unimportance. The highest value (i.e. 1) is the desired answer to be learned by the ANN. In testing mode, the output unit with the higher activation is taken to be the decision of the network as regards the degree of sentence importance. It is because the network can *generalise* and learn based on a relatively small corpus of real-world data that connectionism was chosen for this pragmatic task. A further reason was its ability to eliminate noise and tackle novel data (Carling, 1992).

The Backpropagation learning algorithm was chosen for this three-layer network, because, although relatively slow, it is very reliable and powerful. Still, it is planned to use other network types as well, in order to compare, if nothing else, the related performances with the data at hand. More specifically, BP will be applied in the future on a *recurrent* network with feedback connections keeping track of the context of each sentence, in the form of the feature values of the sentence immediately preceding (Elman, 1988) (Almeida, 1989). Currently, certain aspects of the previous sentences are being considered in the features themselves, e.g. focus change. A feedback loop, though, would make the network take into account the order of the sentences as well, or in other words the concept of time.

A HYBRID ARCHITECTURE

The present ANN operates in a *symbolic* environment consisting of standard Natural Language Processing (NLP) modules for analysis and synthesis [cf. figure 1]. The *analysis* stages provide the content of the input units of the network, which is translated into vectors of values ranging between 0 and 1 by means of an *Encoder*. Most of the features can be represented by discrete values, recording whether or not they are applicable. However, there are some features which need continuous values; e.g. tense needs three values, for past, present and future, respectively. The *synthesis* phases take the sentences selected by the ANN and perform various semantic and syntactic operations on them before generating the abstract of the initial text [cf. (Maybury, 1990)].

A *Hybrid architecture* has been decided upon because adequate symbolic programs already exist that perform standardised lexical, morphological and syntactic analysis satisfactorily. It would have been too difficult to develop the corresponding connectionist machines for these tasks, because there is the problem of *input size variation*, widespread in natural language texts, which is bound to undermine their robustness. As regards *hierarchical structure* and grammar processing, ANNs are not yet controllable enough to perform them (Scholtes, 1990). At any rate, the current emphasis is on the ANN and its ability to discover discriminatory linguistic and pragmatic feature patterns.

THE TEXT CORPUS

In attempting to identify the relevant feature combinations that determine sentence importance, a corpus of real-world texts was developed. More specifically, 55 newspaper articles on world politics and business were analysed manually. Out of these texts 1,100 sentences were selected for training and testing. Their corresponding importance rating was decided manually on the basis of the whole text they belonged to, and *not* on some predetermined features.

Initially, 68 candidate features were identified as possibly relevant to this classification task. These record various aspects of the form, the structure and the development of the text. Example features are: *time*, which records the use of prepositional

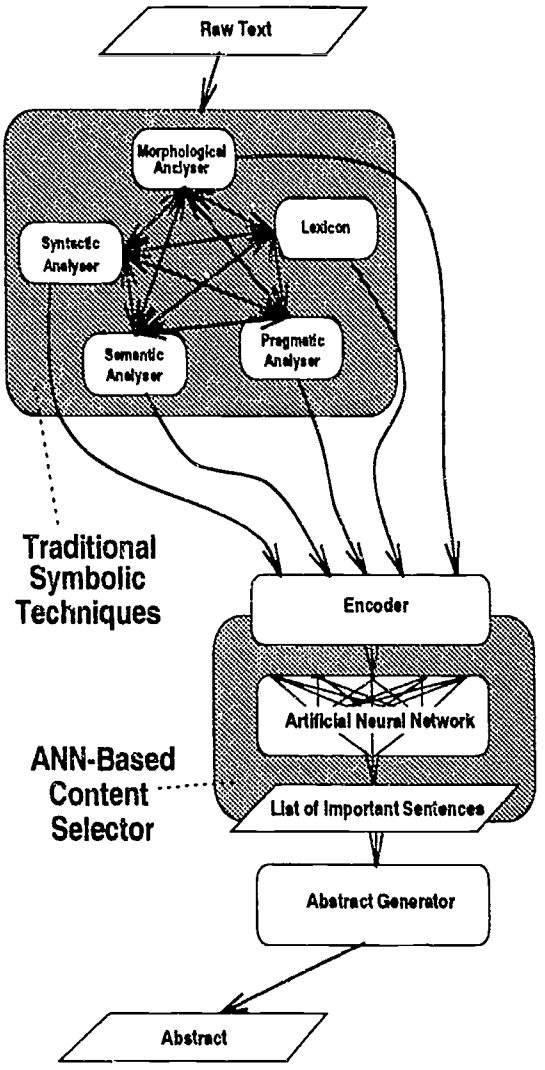

Figure 1: Hybrid System Architecture

phrases, adverbs and subordinate clauses that express a time relation between two or more events and states; *goal*, which refers to the presence of infinitives and other phrases, as well as subordinate clauses, that make known the desired result of a plan; *location* as identified by appropriate prepositional phrases, adverbs and relative clauses; and so on.

As the number of these features is too large for any practical analysis or experimentation to take place [due to the pertinent combinatorial explosion], only *subsets* are employed. Initially, a core set of features was specified, based on intuitions about their potential influence on importance. Experiments were carried out using 6 features, in order to establish the ability of the ANN to perform such a fuzzy task. Later, the relative effect of different sizes of training vs test data was studied, before performing any more serious experiments. Currently, the initial core feature set has been augmented with 6 more features, and all of them are gradually and systematically substituted by other ones from the 68 set. Improvement in performance will hint towards a good feature combination, and vice versa for worsening performance. Thus, non-influential features can be eliminated early on and the focus can be placed on the stronger ones. The aim is to try out all 68 features and probably additional ones discovered in the course of text encoding and the experiments.

To date, 12 features have been used for the encoding of 1,100 sentences by three individuals. Moreover, apart from the aforementioned linguistic and extra-linguistic features, a series of *'meaningless'* features were employed for the encoding of the same sentences. This was done for validation purposes. Example features are: *The second letter of the third word is a vowel, The fourth word has more than one 'S'* or *The first word ends in a consonant.*

As regards the relevant features, the following statistics were observed: In the 1,100 sentence set, *time, goal* and *location* are the features that most clearly imply *importance*. For example, when time is present, it is 59.8% more likely that the corresponding sentence is important. On the other hand, when it is absent, it is 58.4% likely that the sentence is unimportant. The problem is that the remaining features do not present as clear patterns. An example is the feature recording the presence of *anaphors*, i.e. phrases that refer to entities previously mentioned or implied in the text. When an anaphor is present, it is only 49.9% likely that the corresponding sentence is important. When there are no anaphors, the related sentences are important 49.7% of the times. The presence or absence of anaphors cannot be simply discarded as irrelevant to this classification task. This would even be contrary to human intuitions about language use. It is very probable that such features play a discriminatory role in sentence importance as *part of a pattern* of certain other features, rather than individually. It is exactly these patterns, if any, that the ANN has been called to discover.

As for the meaningless features, there is no real correspondence to importance, as expected. Some of them point towards *unimportance*. For example, when the first word starts with a vowel, it is 56.5% likely that the corresponding sentence is unimportant. However, it is obvious that such features depend on chance and could not survive in the long run, when big amounts of test data are involved.

THE EXPERIMENTS

All the experiments reported here were carried out on the PlaNet PDP plattorm (Miyata, 1991). For the time being, only a straightforward BP net-

work is being employed, with a fixed learning rate (0.2) and momentum (0.9). The number of input units depends on how many features are used each time, one unit per feature. The number of hidden units, arranged in a single layer, remains unchanged throughout (30). While, there are two output units, one for importance and one for unimportance, respectively.

The general idea was to start with a few central features to gauge the feasibility of the whole attempt Next, the relationship between the training and the test data size would be explored before any large-scale experimentation takes place. Later, the performance of the network would be monitored as individual features are discarded and others take their place. Improvement in performance would suggest a strong feature or a strong pattern. In order to determine whether the feature itself is relevant or whether it is only relevant if found in specific patterns, different combinations will be tried out containing the controversial feature. Moreover, the influence of adding more features to the existing strong ones will also be traced. No change in performance would mean that the new features may be redundant. In this way, irrelevant features could be eliminated early on, so that the intractability of the combinatorics involved be avoided. More importantly, what will remain in the end should be the most relevant and appropriate features that are necessary and, hopefully, sufficient in determining sentence importance in real texts.

Preliminary Results

Initially, a 100 sentence set was used for both training and testing. The *10-fold cross-validation* technique was adopted, whereby the overall data is divided into 10 different groups of test data (Weiss & Kulikowski, 1991). As a consequence, ten different combinations of 90 training and 10 test sentences were employed in turn for each experiment. In order to check the feasibility of this research only *5 features* were used as input, which gave a *65%* success rate. When a *6th* feature was added, the network performance improved reaching a *69%* success rate. While, when *four more* were added, making up a total of 10 features, the success rate attained an impressive *86%*. This is quite positive given that the great majority of the feature patterns were inconclusive as regards importance. As the data set is too small, most patterns appear just once in the training data and, consequently, the importance judgement may not have been representative of the overall behaviour of the specific pattern, e.g. when it appears in the disparate test data. This is where ANNs seem to outperform the corresponding *statistical analyses*. In general, these experiments showed that there may be a positive correlation between adding more features and and performance improvement. This could mean that the features added were relevant, but it could also be that 5 and 6 features are insufficient for this task, in the first place.

The next step was to scale up the data set from 100 to 1,100 sentences. This time the data was encoded using both relevant and meaningless features. At first, the network was trained with 100 sentences and tested on 900 to 1,000 different ones. The corresponding success rates ranged from *55.9%* to *59.0%*. When the same set-up was used for 12 meaningless features, the success rate lied between 49.2% and 50.2%, which is not better than chance.

The training data set was increased by 100, as a first step towards establishing the relationship between the training and the test data size and its influence on network performance. This time 200 sentences were used for training and 900 for testing. The success rate was *57.0%*, which dropped to 55.7% when 800 test patterns were used. This pointed out the need for further experimentation on this aspect. When the same sentences were used with meaningless features, success varied between 52.2% and 51.2%, respectively.

It was hypothesised that performance depends on who had encoded the training data and who the test data. It so happens that the 100 training sentences used initially were encoded by one individual each time who had only encoded 100, 200 or, at best, 500 sentences in the big test set. Hence, the variation in the success rate. Thus, it was assumed that the mediocre performance of the network was caused by the fact that the data had not been encoded in a uniform way. In order to check this theory, the largest sentence set encoded by a single person, 600 sentences, was used for both training and testing: 100 sentences for training and the remaining 500 for testing. The results were the same as those obtained when 100 sentences from the same big subset had been used for training and 900 for testing, which were also the best: *59.2%* success for the relevant features and 52.2% for the meaningless ones. This means that the success rate depends on the size of the subset of the test data that corresponds to the training data. The bigger the test subset the better the result. Consequently, there must be some discrepancy in the encoding of the data that appears in the experiments as noise.

In order to ensure *wide coverage* in the training data, sentences were compiled from all three individuals, which were then tested on the remaining sentences. A number of experiments were conducted with variable training and test data sizes, in order to establish the effect of the corresponding relation on performance. As shown in figure 2, the success rates are generally low for the sensible features, with the exception of the cases of 250 and 700 training data sizes, which gave the highest rates,

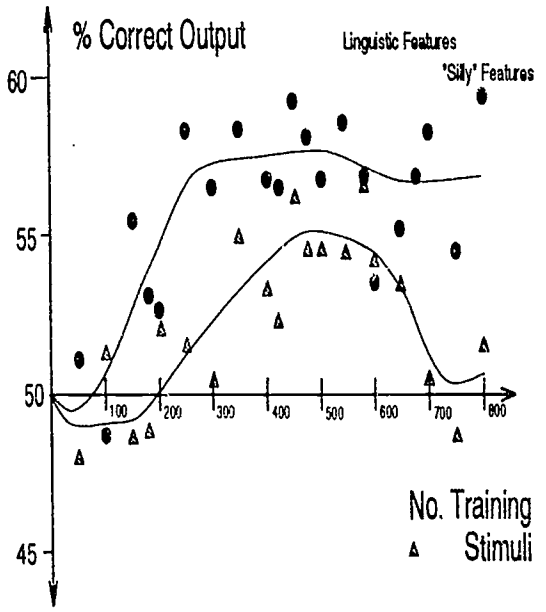

Figure 2: The effect of the training data size on performance

58.4% and 58.2%, respectively. This means that 250 sentences could be sufficient for future training that bears results as good as if 700 sentences were used. If nothing else, the graph suggests that the drop in performance at training sizes bigger than 500 may be due to *over-training*. Despite the mediocrity of these results, the success rates for the meaningless features are even lower, around 50%, and never exceed 54.0%.

DISCUSSION AND FUTURE WORK

The experiments reported in the previous section suggest a number of probable improvements. Firstly, the number of features used so far is still quite small. Twelve features are apparently *insufficient* to allow sentence classification to be accurately formed. Hence, the fact that the success rate is only 8.4% better than random. However, there are many more linguistic and extra-linguistic features from the initial 68 candidate set that remain untried. It is believed that the performance of the network will be improving as more input features are being added.

Secondly, the features employed up to now, or their combinations, may very well be *inappropriate* for the task in hand. This is why different features and feature patterns have to be tried out so that their relative influence in importance determination is established. At least the features already used are not completely irrelevant, as was shown in the comparison of their success rates with those of the meaningless features. Although the network was able to find regularities in the latter, these didn't survive when large data sets were involved. So, it cannot be said that ANNs adapt to data too easily.

At any rate, the relation between the training and the test data sizes has been investigated. Network performance does not significantly improve after a training set of 250 sentences. For this reason it was decided that for all future experiments used to evaluate added and deleted features, the training set size can be kept constant at 500, for example. This should be done in conjunction with the monitoring of variety in the training data as regards the authoring of its encoding. It seems to be the case that different individuals have slightly divergent views regarding the values of individual features and thus the corresponding data is noisy. Additional noise is created by the fact that *diverse text types* are involved: news reports and newspaper articles of varying length, on business and politics. Nevertheless, it is believed that this problem can be overcome by the ANN. This was also one of the reasons for choosing the connectionist paradigm.

Still, sentence encoding needs to be further *standardised* so that feature encoding becomes fully automated in the future. Finally, other types of networks have to be considered later, especially the *recurrent* ones. Thus, objects such as the immediate context of a sentence [the preceding one] could be traced more precisely via feedback layers.

CONCLUSION

In this paper, a feedforward Backpropagation ANN was presented that determines the degree of importance of individual sentences in a text in order to select the ones that will be considered in the generation of the abstract of the text. This network operates in a symbolic environment of conventional analysers and synthesisers. The former provide the values for its input units. Currently, the research focus is on the ANN and its use for the discovery of features and combinations thereof that influence the judgement of importance in real-world texts. Experiments with different, possibly relevant, features are being conducted and the corresponding network performance monitored. Thus, irrelevant features will be gradually eliminated and the fitter ones will be used to guide the development of the symbolic modules of the complete system. For the time being, it can be claimed that ANNs may be capable and appropriate to take pragmatic decisions, because there is ample space for improvement as regards the experimental results collected so far.

References

Almeida, L. B. (1989). Backpropagation in Nonfeedforward Networks. In I. Aleksander (Ed.), *Neural*

Computing Architectures. Cambridge, MA: MIT Press.

Appelt, D. E. & Hobbs, J. R. & Bear, J. & Israel, D. & Tyson, M. (1993). FASTUS: A Finite-state Processor for Information Extraction from Real-world Text. In *Proc. 13th Int. Joint Conf. On Artificial Intelligence (IJCAI-93)*, August 28-September 3, Chabéry: Morgan Kaufmann.

Carling, A. (1992). *Introducing Neural Networks*. Wilmslow, Cheshire, U.K.: Sigma Press.

Elman, J. L. (1988). *Finding Structure in Time*. Technical Report CRL TR-8801, Center for Research on Language, University of California, San Diego.

Lucas, N. & Nishina, K. & Akiba, T. & Suresh, K. G. (1993). *Discourse Analysis of scientific textbooks in Japanese: a tool for producing automatic summaries*. Technical Report 93TR-0004, Dept. of Computer Science, Tokyo Institute of Technology, Tokyo, Japan.

Mauldin, M. L., Carbonell, J. G. & Thomason, R. H. (1987). Beyond the Keyword Barrier: Knowledge-Based Information Retrieval. In *Proc. 29th Annual Conference of the National Federation of Abstracting and Information Services*. Also appeared in *Information Services and Use*, Vol. 7.

Maybury, M. T. (1990). *Planning Multisentential English Text Using Communication Acts*. PhD thesis, Cambridge University Computer Laboratory, September.

Miyata, Y. (1991). *A User's Guide to PlaNet Version 5.6. A Tool for constructing, running, and looking into a PDP network*. Copyright 1987, 1988, 1989, 1990 by Yoshiro Miyata.

Rumelhart, D. E. & Hinton, G. E. & Williams, R. J. (1986). Learning Internal Representations by Error Propagation. In D. E. Rumelhart & J. L. McClelland (Ed.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition; Vol. 1: Foundations*. Cambridge, MA: MIT Press.

Scholtes, J. C. (1990). *Neural Networks In Natural Language Processing And Information Retrieval*. PhD thesis, University of Amsterdam, The Netherlands.

Weiss, S. & Kulikowski, C. (1991). *Computer Systems that Learn*. San Mateo, CA: Morgan Kaufmann.

Young, S. R. & Hayes, P. J. (1985). TESS: A Telex Classifying and Summarisation System. In *Proc. 2nd IEEE Conference on AI Applications*.