NOUN PHRASE ANALYSIS USING THE IBM 650 COMPUTER

ARISTOTELIS D. STATHACOPOULOS
Graduate Student in Electrical Engineering

At present, machine translation is accomplished by a word-for-word translating process, including some idiomatic phrases treated as independent units and translated as such. It is easily apparent that translation of this kind carries the desired information, but it requires special effort by the reader in order to be understood. The output of such a translating device will appear in the source-language word order and it will include multiple translations for words with . several meanings in the target language. In other words, such an output is not in conventional English and requires polishing and rewriting before it can compare with the output of a good human translator.

Rearrangement of word order and selection of the contextual meaning of a word constitute two of the main problems of machine translation. A device which will overcome these difficulties should be able to perform certain logical operations which the mind of the human translator performs unconsciously.

This fact suggests the use of a modern computer with permanent stored program to analyze the source-language text and perform the changes required in the word-for-word equivalent target-language translation.

One particular problem is recognition of the word series consisting of article, adjectives, and final noun, called a noun phrase, which replaces the simple noun in more complex sentence structure. The following discussion describes a method for recognizing and analyzing a noun phrase in any sentence.

A noun phrase consists of articles, adjectives, and nouns. However, many of the constituents of a noun phrase may belong to more than one grammatical class. These words must be assigned to their correct class on the basis of context and translated accordingly. If this is to be done by a computer, each word must be accompanied by a class tag operated on by the computer program as it performs the desired logical operation while analyzing a group of words.

Binary numbers were selected as tags and the intersection sum of logical algebra was selected as the main operation for the program. In taking the intersection sum of two binary numbers the answer will have a *one* whenever we add two *ones* in the same column, while it will have zeros in any other case. The example below illustrates what is meant by the intersection sum.

```
11001
          ← binary numbers
10010
-----
10000 ← intersection sum
```

During the following discussion, by "addition" we mean the intersection sum.

The tags selected for this analysis are the following:

article	110001
adjective	100001
noun	100000

These tags have the following properties of logical algebra:

art ∩art. =art. art. ∩adj. = adj. art. ∩noum = noun
adj. ∩adj. = adj. adj. ∩ noun = noun
noun∩noun = noun

The symbol ∩ indicates the intersection sum.

Each word will be represented by one tag for each grammatical class to which it belongs. Thus, words belonging to several grammatical classes will be represented by a series of tags in a predetermined order. For example, the word "book" will have the following series of tags:

100001 100000 111111

The tag 111111 is the tag assigned to any verb.

The following description outlines the essential features of the flow chart of the program as shown in Fig. 1, which provides for a noun phrase of up to six words.

The first tag of the first word is examined to determine whether it marks the beginning off a noun phrase or not. If not, a further analysis follows to determine the nature of the tag. However, if the first tag indicates the beginning of a noun phrase, the routine continues as follows: The first tags of the next five words are added consecutively to this tag, and after every addition, the result is tested to determine whether we have a noun phrase or not. If the first tags fail to indicate a noun phrase, the second tags are examined until a noun phrase tag is formed.

The following example of a three-word noun phrase will demonstrate the performance of the program:

The Australian team appeared.

```
110001   100001   100001   111111
000000   000000   100000   000000
```

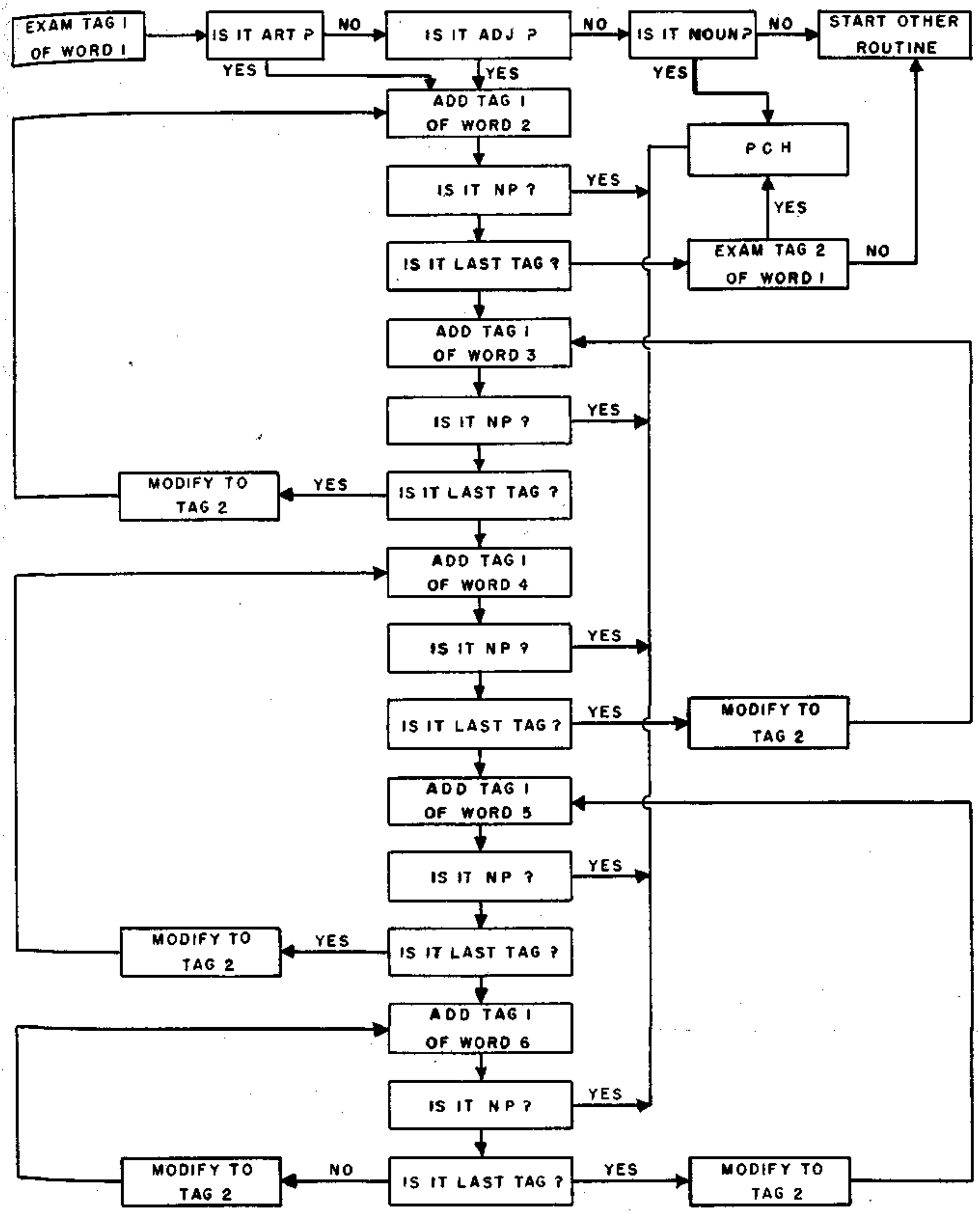

FIG. 1. FLOW CHART

Initially, the first tag for the word "the" is analyzed, and since it is an article tag, it marks the beginning of a noun phrase. Following this, the first tag of the word "Australian" is added to the previous tag and the result is 100001 which is not the noun phrase tag. Consecutively, in the same fashion, the first tags for the words "team" and "appeared" are added, but still these tags fail to form a noun phrase tag, 100000. Then the first tag of the word "appeared" is replaced by the second tag and a new intersection operation is performed. Still no noun phrase is obtained. Then "appeared" is dropped from the process. Next the first tag of the word "team" is replaced by its second tag and a new intersection sum is obtained. The final result is 100000, which indicates the end of the noun phrase.

At this point, a card will be punched by the machine with the tags 110001, 100001, 100000, which are the correct tags of the analyzed noun phrase "The Australian team." In this process the noun phrase was identified and "team" was found to belong to the noun class in this context.

Since the IBM 650 computer is a bi-quinary machine and is not capable of performing intersection summations, a special subroutine has been included in the program which is designed to transform the normal summation result to the corresponding intersection sum.

The program is able to determine the beginning and the end of the noun phrase and assign the correct grammatical class to each constituent. It also demonstrates the possibility of applying mathematical operations to sentence analysis and the use of a computer to carry out the process.

A complete program of this kind, constructed for analyzing sentences of the source language, could then be incorporated into a word-for-word translation and work on the tags of the source-language words. It could be expanded further to be capable of transforming the source-language analysis results to the corresponding target-language analysis and thus improve the accuracy, understandability, and readability of the final translation.