A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text

Kenneth Ward Church
Bell Laboratories
600 Mountain Ave.
Murray Hill, N.J., USA
201-582-5325
alice!kwc

It is well-known that part of speech depends on context. The word ‘‘table,’’ for example, can be a verb in some contexts (e.g., ‘‘He will table the motion’’) and a noun in others (e.g., ‘‘The table is ready’’). A program has been written which tags each word in an input sentence with the most likely part of speech. The program produces the following output for the two ‘‘table’’ sentences just mentioned:

- He/PPS will/MD table/VB the/AT motion/NN ./.
- The/AT table/NN is/BEZ ready/JJ ./.

(PPS = subject pronoun; MD = modal; VB = verb (no inflection); AT = article; NN = noun; BEZ = present 3rd sg form of ‘‘to be’’; JJ = adjective; notation is borrowed from [Francis and Kucera, pp. 6-8])

Part of speech tagging is an important practical problem with potential applications in many areas including speech synthesis, speech recognition, spelling correction, proof-reading, query answering, machine translation and searching large text data bases (e.g., patents, newspapers). The author is particularly interested in speech synthesis applications, where it is clear that pronunciation sometimes depends on part of speech. Consider the following three examples where pronunciation depends on part of speech. First, there are words like ‘‘wind’’ where the noun has a different vowel than the verb. That is, the noun ‘‘wind’’ has a short vowel as in ‘‘the wind is strong,’’ whereas the verb ‘‘wind’’ has a long vowel as in ‘‘Don’t forget to wind your watch.’’ Secondly, the pronoun ‘‘that’’ is stressed as in ‘‘Did you see THAT?’’ unlike the complementizer ‘‘that,’’ as in ‘‘It is a shame that he’s leaving.’’ Thirdly, note the difference between ‘‘oily FLUID’’ and ‘‘TRANSMISSION fluid’’; as a general rule, an adjective-noun sequence such as ‘‘oily FLUID’’ is typically stressed on the right whereas a noun-noun sequence such as ‘‘TRANSMISSION fluid’’ is typically stressed on the left. These are but three of the many constructions which would sound more natural if the synthesizer had access to accurate part of speech information.

Perhaps the most important application of tagging programs is as a tool for future research. A number of large projects such as [Cobuild] have recently been collecting large corpora (10-1000 million words) in order to better describe how language is actually used in practice:

> ‘‘For the first time, a dictionary has been compiled by the thorough examination of representative group of English texts, spoken and written, running to many millions of words. This means that in addition to all the tools of the conventional dictionary makers... the dictionary is based on hard, measureable evidence.’’ [Cobuild, p. xv]

It is likely that there will be more and more research projects collecting larger and larger corpora. A reliable parts program might greatly enhance the value of these corpora to many of these researchers.

The program uses a linear time dynamic programming algorithm to find an assignment of parts of speech to words that optimizes the product of (a) lexical probabilities (probability of observing part of speech i given word j), and (b) contextual probabilities (probability of observing part of speech i given k previous parts of speech). Probability estimates were obtained by training on the Tagged Brown Corpus [Francis and Kucera], a corpus of approximately 1,000,000 words with part of speech tags assigned laboriously by hand over many years.

Program performance is encouraging (95-99% ‘‘correct’’, depending on the definition of ‘‘correct’’). A small 400 word sample is presented in the Appendix, and is judged to be 99.5% correct. It is surprising that a local ‘‘bottom-up’’ approach can perform so well. Most errors are attributable to defects in the lexicon; remarkably few errors are related to the inadequacies of the extremely over-simplified grammar (a trigram model). Apparently, ‘‘long distance’’ dependences are not very important, at

least most of the time.

One might have thought that ngram models weren't adequate for the task since it is well-known that they are inadequate for determining grammaticality:

> ''We find that no finite-state Markov process that produces symbols with transition from state to state can serve as an English grammar. Furthermore, the particular subclass of such processes that produce n-order statistical approximations to English do not come closer, with increasing *n*, to matching the output of an English grammar.'' [Chomsky, p. 113]

Chomsky's conclusion was based on the observation that constructions such as:

- If S_1 then S_2.
- Either S_3, or S_4.
- The man who said that S_5, is arriving today.

have long distance dependencies that span across any fixed length window *n*. Thus, ngram models are clearly inadequate for many natural language applications. However, for the tagging application, the ngram approximation may be acceptable since long distance dependencies do not seem to be very important.

Statistical ngram models were quite popular in the 1950s, and have been regaining popularity over the past few years. The IBM speech group is perhaps the strongest advocate of ngram methods, especially in other applications such as speech recognition. Robert Mercer (private communication, 1982) has experimented with the tagging application, using a restricted corpus (laser patents) and small vocabulary (1000 words). Another group of researchers working in Lancaster around the same time, Leech, Garside and Atwell, also found ngram models highly effective; they report 96.7% success in automatically tagging the LOB Corpus, using a bigram model modified with heuristics to cope with more important trigrams. The present work developed independently from the LOB project.

1. How Hard is Lexical Ambiguity?

Many people who have not worked in computational linguistics have a strong intuition that lexical ambiguity is usually not much of a problem. It is commonly believed that most words have just one part of speech, and that the few exceptions such as ''table'' are easily disambiguated by context in most cases. In contrast, most experts in computational linguists have found lexical ambiguity to be a major issue; it is said that practically any content word can be used as a noun, verb or adjective,[1] and that local context is not always adequate to disambiguate. Introductory texts are full of ambiguous sentences such as

- Time flies like an arrow.
- Flying planes can be dangerous.

where no amount of syntactic parsing will help. These examples are generally taken to indicate that the parser must allow for multiple possibilities and that grammar formalisms such as LR(k) are inadequate for natural language since these formalisms cannot cope with ambiguity. This argument was behind a large set of objections to Marcus' ''LR(k)-like'' Deterministic Parser.

Although it is clear that an expert in computational linguistics can dream up arbitrarily hard sentences, it may be, as Marcus suggested, that most texts are not very hard in practice. Recall that Marcus hypothesized most decisions can be resolved by the parser within a small window (i.e., three buffer cells), and there are only a few problematic cases where the parser becomes confused. He called these confusing cases ''garden paths,'' by analogy with the famous example:

- The horse raced past the barn fell.

With just a few exceptions such as these ''garden paths,'' Marcus assumes, there is almost always a unique ''best'' interpretation which can be found with very limited resources. The proposed stochastic approach is largely compatible with this; the proposed approach assumes that it is almost always sufficient to assign each word a unique ''best'' part of speech (and this can be

1. From an information theory point of view, one can quantity ambiguity in bits. In the case of the Brown Tagged Corpus, the lexical entropy, the conditional entropy of the part of speech given the word is about 0.25 bits per part of speech. This is considerably smaller than the contextual entropy, the conditional entropy of the part of speech given the next two parts of speech. This entropy is estimated to be about 2 bits per part of speech.

accomplished with a very efficient linear time dynamic programming algorithm). After reading introductory discussions of ‘‘Flying planes can be dangerous,’’ one might have expected that lexical ambiguity was so pervasive that it would be hopeless to try to assign just one part of speech to each word and in just one linear time pass over the input words.

2. Lexical Disambiguation Rules

However, the proposed stochastic method is considerably simpler than what Marcus had in mind. His thesis parser used considerably more syntax than the proposed stochastic method. Consider the following pair described in [Marcus]:

- Have/VB [the students who missed the exam] TAKE the exam today. (imperative)
- Have/AUX [the students who missed the exam] TAKEN the exam today? (question)

where it appears that the parser needs to look past an arbitrarily long noun phrase in order to correctly analyze ‘‘have,’’ which could be either a tenseless main verb (imperative) or a tensed auxiliary verb (question). Marcus’ rather unusual example can no longer be handled by Fidditch, a more recent Marcus-style parser with very large coverage. In order to obtain such large coverage, Fidditch has had to take a more robust/modest view of lexical disambiguation. Whereas Marcus’ Parsifal program distinguished patterns such as ‘‘have NP tenseless’’ and ‘‘have NP past-participle,’’ most of Fidditch’s diagnostic rules are less ambitious and look only for the start of a noun phrase and do not attempt to look past an arbitrarily long noun phrase. For example, Fidditch has the following lexical disambiguation rule:

- (defrule n+prep!
  " > [**n+prep] != n [npstarters]")

which says that a preposition is more likely than a noun before a noun phrase. More precisely, the rule says that if a noun/preposition ambiguous word (e.g., ‘‘out’’) is followed by something that starts a noun phrase (e.g., a determiner), then rule out the noun possibility. This type of lexical diagnostic rule can be captured with bigram and trigram statistics; it turns out that the sequence ...preposition determiner... is much more common in the Brown Corpus (43924 observations) than the sequence ...noun determiner... (1135 observations). Most lexical disambiguation rules in Fidditch can be reformulated in terms of bigram and trigram statistics in this way. Moreover, it is worth doing so, because bigram and trigram statistics are much easier to obtain than Fidditch-type disambiguation rules, which are extremely tedious to program, test and debug.

In addition, the proposed stochastic approach can naturally take advantage of lexical probabilities in a way that is not easy to capture with parsers that do not make use of frequency information. Consider, for example, the word ‘‘see,’’ which is almost always a verb, but does have an archaic nominal usage as in ‘‘the Holy See.’’ For practical purposes, ‘‘see’’ should not be considered noun/verb ambiguous in the same sense as truly ambiguous words like ‘‘program,’’ ‘‘house’’ and ‘‘wind’’; the nominal usage of ‘‘see’’ is possible, but not likely.

If every possibility in the dictionary must be given equal weight, parsing is very difficult. Dictionaries tend to focus on what is possible, not on what is likely. Consider the trivial sentence, ‘‘I see a bird.’’ For all practical purposes, every word in the sentence is unambiguous. According to [Francis and Kucera], the word ‘‘I’’ appears as a pronoun (PPLS) in 5837 out of 5838 observations (˜100%), ‘‘see’’ appears as a verb in 771 out of 772 observations (˜100%), ‘‘a’’ appears as an article in 23013 out of 23019 observations (˜100%) and ‘‘bird’’ appears as a noun in 26 out of 26 observations (˜100%). However, according to Webster’s Seventh New Collegiate Dictionary, every word is ambiguous. In addition to the desired assignments of tags, the first three words are listed as nouns and the last as an intransitive verb. One might hope that these spurious assignments could be ruled out by the parser as syntactically ill-formed. Unfortunately, this is unlikely to work. If the parser is going to accept noun phrases of the form:

- [NP [N city] [N school] [N committee] [N meeting]]

then it can’t rule out

- [NP [N I] [N see] [N a] [N bird]]

Similarly, the parser probably also has to accept ‘‘bird’’ as an intransitive verb, since there is nothing syntactically wrong with:

- [S [NP [N I] [N see] [N a]] [VP [V bird]]]

These part of speech assignments aren't wrong; they are just extremely improbable.

3. The Proposed Method

Consider once again the sentence, ''I see a bird.'' The problem is to find an assignment of parts of speech to words that optimizes both lexical and contextual probabilities, both of which are estimated from the Tagged Brown Corpus. The lexical probabilities are estimated from the following frequencies:

Word	**Parts of Speech**			
I	PPSS	5837	NP	1
see	VB	771	UH	1
a	AT	23013	IN (French)	6
bird	NN	26		

(PPSS = pronoun; NP = proper noun; VB = verb; UH = interjection; IN = preposition; AT = article; NN = noun)

The lexical probabilities are estimated in the obvious way. For example, the probability that ''I'' is a pronoun, Prob(PPSS|''I''), is estimated as the freq(PPSS|''I'')/freq(''I'') or 5837/5838. The probability that ''see'' is a verb is estimated to be 771/772. The other lexical probability estimates follow the same pattern.

The contextual probability, the probability of observing part of speech X given the following two parts of speech Y and Z, is estimated by dividing the trigram frequency XYZ by the bigram frequency YZ. Thus, for example, the probability of observing a verb before an article and a noun is estimated to be the ratio of the freq(VB, AT, NN) over the freq(AT, NN) or 3412/53091 = 0.064. The probability of observing a noun in the same context is estimated as the ratio of freq(NN, AT, NN) over 53091 or 629/53091 = 0.01. The other contextual probability estimates follow the same pattern.

A search is performed in order to find the assignment of part of speech tags to words that optimizes the product of the lexical and contextual probabilities. Conceptually, the search enumerates all possible assignments of parts of speech to input words. In this case, there are four input words, three of which are two ways ambiguous, producing a set of 2*2*2*1=8 possible assignments of parts of speech to input words:

I	see	a	bird
PPSS	VB	AT	NN
PPSS	VB	IN	NN
PPSS	UH	AT	NN
PPSS	UH	IN	NN
NP	VB	AT	NN
NP	VB	IN	NN
NP	UH	AT	NN
NP	UH	IN	NN

Each of the eight sequences are then scored by the product of the lexical probabilities and the contextual probabilities, and the best sequence is selected. In this case, the first sequence is by far the best.

In fact, it is not necessary to enumerate all possible assignments because the scoring function cannot see more than two words away. In other words, in the process of enumerating part of speech sequences, it is possible in some cases to know that some sequence cannot possibly compete with another and can therefore be abandoned. Because of this fact, only O(n) paths will be enumerated. Let us illustrate this optimization with an example:

Find all assignments of parts of speech to ''bird'' and score the partial sequence. Henceforth, all scores are to be interpreted as log probabilities.

(-4.848072 "NN")

Find all assignments of parts of speech to ''a'' and score. At this point, there are two paths:

(-7.4453945 "AT" "NN")
(-15.01957 "IN" "NN")

Now, find assignments of ''see'' and score. At this point, the number of paths seem to be growing exponentially.

(-10.1914 "VB" "AT" "NN")
(-18.54318 "VB" "IN" "NN")
(-29.974142 "UH" "AT" "NN")
(-36.53299 "UH" "IN" "NN")

Now, find assignments of ''I'' and score. Note, however, that it is no longer necessary to hypothesize that ''a'' might be a French

preposition IN because all four paths, PPSS VB IN NN, NN VB IN NN, PPSS UH IN NN and NP UH AT NN score less well than some other path and there is no way that any additional input could make any difference. In particular, the path, PPSS VB IN NN scores less well than the path PPSS VB AT NN, and additional input will not help PPSS VB IN NN because the contextual scoring function has a limited window of three parts of speech, and that is not enough to see past the existing PPSS and VB.

(-12.927581 "PPSS" "VB" "AT" "NN")
(-24.177242 "NP" "VB" "AT" "NN")
(-35.667458 "PPSS" "UH" "AT" "NN")
(-44.33943 "NP" "UH" "AT" "NN")

The search continues two more iterations, assuming blank parts of speech for words out of range.

(-13.262333 "" "PPSS" "VB" "AT" "NN")
(-26.5196 "" "NP" "VB" "AT" "NN")

Finally, the result is: PPSS VB AT NN.

(-12.262333 "" "" "PPSS" "VB" "AT" "NN")

The final result is: I/PPSS see/VB a/AT bird/NN.

A slightly more interesting example is: ‘‘Can they can cans.’’

cans
(-5.456845 "NNS")

can
(-12.603266 "NN" "NNS")
(-15.935471 "VB" "NNS")
(-15.946739 "MD" "NNS")

they
(-18.02618 "PPSS" "MD" "NNS")
(-18.779934 "PPSS" "VB" "NNS")
(-21.411636 "PPSS" "NN" "NNS")

can
(-21.766554 "MD" "PPSS" "VB" "NNS")
(-26.45485 "NN" "PPSS" "MD" "NNS")
(-28.306572 "VB" "PPSS" "MD" "NNS")

(-21.932137 "" "MD" "PPSS" "VB" "NNS")
(-30.170452 "" "VB" "PPSS" "MD" "NNS")
(-31.453785 "" "NN" "PPSS" "MD" "NNS")

And the result is: Can/MD they/PPSS can/VB cans/NNS

(-20.932137 "" "" "MD" "PPSS" "VB" "NNS")

4. Parsing Simple Non-Recursive Noun Phrases Stochastically

Similar stochastic methods have been applied to locate simple noun phrases with very high accuracy. The program inserts brackets into a sequence of parts of speech, producing output such as:

[A/AT former/AP top/NN aide/NN] to/IN [Attorney/NP General/NP Edwin/NP Meese/NP] interceded/VBD to/TO extend/VB [an/AT aircraft/NN company/NN [business/NN] with/IN [a/AT lobbyist/NN] [who/WPS] worked/VBD for/IN [the/AT defense/NN contractor/NN] ,/, according/IN to/IN [a/AT published/VBN report/NN] ./.

The proposed method is a stochastic analog of precedence parsing. Recall that precedence parsing makes use of a table that says whether to insert an open or close bracket between any two categories (terminal or nonterminal). The proposed method makes use of a table that gives the probabilities of an open and close bracket between all pairs of parts of speech. A sample is shown below for the five parts of speech: AT (article), NN (singular noun), NNS (non-singular noun), VB (uninflected verb), IN (preposition). The table says, for example, that there is no chance of starting a noun phrases after an article (all five entries on the AT row are 0) and that there is a large probability of starting a noun phrase between a verb and an noun (the entry in (VB, AT) is 1.0).

Probability of Starting a Noun Phrase

	AT	**NN**	**NNS**	**VB**	**IN**
AT	0	0	0	0	0
NN	.99	.01	0	0	0
NNS	1.0	.02	.11	0	0
VB	1.0	1.0	1.0	0	0
IN	1.0	1.0	1.0	0	0

Probability of Ending a Noun Phrase

	AT	NN	NNS	VB	IN
AT	0	0	0	0	0
NN	1.0	.01	0	0	1.0
NNS	1.0	.02	.11	1.0	1.0
VB	0	0	0	0	0
IN	0	0	0	0	.02

These probabilities were estimated from about 40,000 words (11,000 noun phrases) of training material selected from the Brown Corpus. The training material was parsed into noun phrases by laborious semi-automatic means (with considerable help from Eva Ejerhed). It took about a man-week to prepare the training material.

The stochastic parser is given a sequence of parts of speech as input and is asked to insert brackets corresponding to the beginning and end of noun phrases. Conceptually, the parser enumerates all possible parsings of the input and scores each of them by the precedence probabilities. Consider, for example, the input sequence: NN VB. There are 5 possible ways to bracket this sequence (assuming no recursion):

- NN VB
- [NN] VB
- [NN VB]
- [NN] [VB]
- NN [VB]

Each of these parsings is scored by multiplying 6 precedence probabilities, the probability of an open/close bracket appearing (or not appearing) in any one of the three positions (before the NN, after the NN or after the VB). The parsing with the highest score is returned as output.

A small sample of the output is given in the appendix. The method works remarkably well considering how simple it is. There is some tendency to underestimate the number of brackets and run two noun phrases together as in [NP *the time Fairchild*]. The proposed method omitted only 5 of 243 noun phrase brackets in the appendix.

5. Smoothing Issues

Some of the probabilities are very hard to estimate by direct counting because of Zipf's Law (frequency is roughly proportional to inverse rank). Consider, for example, the lexical probabilities. We need to estimate how often each word appears with each part of speech. Unfortunately, because of Zipf's Law, no matter how much text we look at, there will always be a large tail of words that appear only a few times. In the Brown Corpus, for example, 40,000 words appear five times or less. If a word such as ''yawn'' appears once as a noun and once as a verb, what is the probability that it can be an adjective? It is impossible to say without more information. Fortunately, conventional dictionaries can help alleviate this problem to some extent. We add one to the frequency count of possibilities in the dictionary. For example, ''yawn'' happens to be listed in our dictionary as noun/verb ambiguous. Thus, we smooth the frequency counts obtained from the Brown Corpus by adding one to both possibilities. In this case, the probabilities remain unchanged. Both before and after smoothing, we estimate ''yawn'' to be a noun 50% of the time, and a verb the rest. There is no chance that ''yawn'' is an adjective.

In some other cases, smoothing makes a big difference. Consider the word ''cans.'' This word appears 5 times as a plural noun and never as a verb in the Brown Corpus. The lexicon (and its morphological routines), fortunately, give both possibilities. Thus, the revised estimate is that ''cans'' appears 6/7 times as a plural noun and 1/7 times as a verb.

Proper nouns and capitalized words are particularly problematic; some capitalized words are proper nouns and some are not. Estimates from the Brown Corpus can be misleading. For example, the capitalized word ''Acts'' is found twice in the Brown Corpus, both times as a proper noun (in a title). It would be a mistake to infer from this evidence that the word ''Acts'' is always a proper noun. For this reason, capitalized words with small frequency counts (< 20) were thrown out of the lexicon.

There are other problems with capitalized words. Consider, for example, a sentence beginning with the capitalized word ''Fall''; what is the probability that it is a proper noun (i.e., a surname)? Estimates from the Brown Corpus are of little help here since ''Fall'' never appears as a capitalized word and it never appears as a proper noun. Two steps were taken to alleviate this problem. First, the frequency estimates for

‘‘Fall’’ are computed from the estimates for ‘‘fall’’ plus 1 for the proper noun possibility. Thus, ‘‘Fall’’ has frequency estimates of: ((1 . "NP") (1 . "JJ") (65 . "VB") (72 . "NN")) because ‘‘fall’’ has the estimates of: ((1 . "JJ") (65 . "VB") (72 . "NN")). Secondly, a prepass was introduced which labels words as proper nouns if they are ‘‘adjacent to’’ other capitalized words (e.g., ‘‘White House,’’ ‘‘State of the Union’’) or if they appear several times in a discourse and are always capitalized.

The lexical probabilities are not the only probabilities that require smoothing. Contextual frequencies also seem to follow Zipf’s Law. That is, for the set of all sequences of three parts of speech, we have plotted the frequency of the sequence against its rank on log log paper and observed the classic (approximately) linear relationship and slope of (almost) -1. It is clear that the contextual frequencies require smoothing. Zeros should be avoided.

6. Conclusion

A stochastic part of speech program and noun phrase parser has been presented. Performance is very encouraging as can be seen from the Appendix.

References

Chomsky, N., ‘‘Three Models for the Description of Language,’’ IRE Transactions on Information Theory, vol. IT-2, Proceedings of the Symposium on Information Theory, 1956.

‘‘Collins Cobuild English Language Dictionary,’’ William Collins Sons & Co Ltd, 1987.

Ejerhed, E., ‘‘Finding Clauses in Unrestricted Text by Stochastic and Finitary Methods,’’ abstracted submitted to this conference.

Francis, W., and Kucera, H., ‘‘Frequency Analysis of English Usage,’’ Houghton Mifflin Company, Boston, 1982.

Leech, G., Garside, R., Atwell, E., ‘‘The Automatic Grammatical Tagging of the LOB Corpus,’’ ICAME News 7, 13-33, 1983.

Marcus, M., ‘‘A Theory of Syntactic Recognition for Natural Language,’’ MIT Press, Cambridge, Massachusetts, 1980.

‘‘Webster’s Seventh New Collegiate Dictionary,’’ Merriam Company, Springfield, Massachusetts, 1972.

Appendix: Sample Results

The following story was distributed over the AP during the week of May 26, 1987. There are just two tagging errors which are indicated with ‘‘***’’. There are five missing brackets which are indicated as ‘‘*[’’ or ‘‘*]’’. Words with a second NP tag were identified as proper nouns in a prepass.

[A/AT former/AP top/NN aide/NN] to/IN [Attorney/NP/NP General/NP/NP Edwin/NP/NP Meese/NP/NP] interceded/VBD to/TO extend/VB [an/AT aircraft/NN company/NN ’s/$ government/NN contract/NN] ,/, then/RB went/VBD into/IN [business/NN] with/IN [a/AT lobbyist/NN] [who/WPS] worked/VBD for/IN [the/AT defense/NN contractor/NN] ,/, according/IN to/IN [a/AT published/VBN report/NN] ./.

[James/NP/NP E/NP./NP Jenkins/NP/NP] ,/, [a/AT one-time/JJ senior/JJ deputy/NN] to/IN [Meese/NP/NP] ,/, joined/VBD [the/AT board/NN] of/IN [directors/NNS] of/IN [Transworld/NP/NP Group/NP/NP Ltd/NP./NP] on/IN [April/NP/NP 28/CD] ,/, [1984/CD] ,/, [the/AT Chicago/NP/NP Tribune/NP/NP] reported/VBD in/IN [its/PP$ Tuesday/NR editions/NNS] ./.

[The/AT principal/JJ figure/NN] in/IN [Transworld/NP/NP] was/BEDZ [Richard/NP/NP Millman/NP/NP] ,/, [a/AT lobbyist/NN] for/IN [Fairchild/NP/NP Industries/NP/NP Inc/NP./NP] ,/, [a/AT Virginia/NP/NP defense/NN contractor/NN] ,/, [the/AT Tribune/NP/NP] said/VBD ./.

[A/AT federal/JJ grand/JJ jury/NN] is/BEZ investigating/VBG [the/AT Fairchild/NP/NP transaction/NN] and/CC [other/AP actions/NNS] of/IN [Meese/NP/NP] and/CC [former/AP White/NP/NP House/NP/NP aide/NN Lyn/NP/NP Nofziger/NP/NP] in/IN [connection/NN] with/IN [Wedtech/NP/NP Corp/NP./NP] ,/, [a/AT New/NP/NP York/NP/NP defense/NN company/NN] [that/WPS] received/VBD [$250/CD million/CD] in/IN [government/NN contracts/NNS] issued/VBN without/IN [competitive/JJ bidding/NN] during/IN [the/AT Reagan/NP/NP administration/NN] ./.

[Jenkins/NP/NP] left/VBD [the/AT White/NP/NP House/NP/NP] in/IN [1984/CD] ,/, and/CC joined/VBD [Wedtech/NP/NP] as/CS [its/PP$ director/NN] of/IN [marketing/NN *]*[two/CD years/NNS] later/RBR ./.

[Deborah/NP/NP Tucker/NP/NP] ,/, [a/AT spokeswoman/NN] for/IN [Fairchild/NP/NP] ,/, said/VBD [Friday/NR] that/CS [the/AT company/NN] had/HVD been/BEN contacted/VBN by/IN [the/AT office/NN] of/IN [independent/JJ counsel/NN James/NP/NP McKay/NP/NP] and/CC [subpoenas/NNS] had/HVD been/BEN served/VBN on/IN [Fairchild/NP/NP] ./.

[Tucker/NP/NP] said/VBD [the/AT investigation/NN] involving/IN [Fairchild/NP/NP] had/HVD been/BEN going/VBG on/IN [a/AT number/NN] of/IN [weeks/NNS] and/CC predates/VBZ [last/AP week/NN 's/$ expansion/NN] of/IN [McKay/NP/NP 's/$ investigation/NN] to/TO include/VB [Meese/NP/NP] ./.

[The/AT company/NN] is/BEZ cooperating/VBG in/IN [the/AT investigation/NN] ,/, [Tucker/NP/NP] said/VBD ./.

[A/AT source/NN *] close/NN***] to/IN [McKay/NP/NP] said/VBD [last/AP week/NN] that/CS [Meese/NP/NP] isn't/BEZ* under/IN [criminal/JJ investigation/NN] in/IN [the/AT Fairchild/NP/NP matter/NN] ,/, but/RB is/BEZ [a/AT witness/NN] ./.

[The/NP Tribune/NP/NP] said/VBD [Millman/NP/NP] ,/, acting/VBG as/CS [a/AT lobbyist/NN] for/IN [the/AT Chantilly/NP/NP] ,/, [Va/NP.-based/NP company/NN] ,/, went/VBD to/TO see/VB [Jenkins/NP/NP] in/IN [1982/CD] and/CC urged/VBD [him/PPO] and/CC [Meese/NP/NP] to/TO encourage/VB [the/AT Air/NP/NP Force/NP/NP] to/TO extend/VB [the/AT production/NN] of/IN [Fairchild/NP/NP 's/$ A-10/NP bomber/NN] for/IN [a/AT year/NN] ./.

[Millman/NP/NP] said/VBD there/RB was/BEDZ [a/AT lucrative/JJ market/NN] in/IN [Third/NP/NP World/NP/NP countries/NNS] ,/, but/CC that/CS [Fairchild/NP/NP 's/$ chances/NNS] would/MD be/BE limited/VBN if/CS [the/AT Air/NP/NP Force/NP/NP] was/BEDZ not/* producing/VBG [the/AT plane/NN] ./.

[The/AT Air/NP/NP Force/NP/NP] had/HVD decided/VBN to/TO discontinue/VB [production/NN] of/IN [the/AT A-10/NP] ,/, [a/AT 1960s-era/CD ground-support/NN attack/NN bomber/NN] ,/, at/IN [the/AT time/NN *]*[Fairchild/NP/NP] was/BEDZ hoping/VBG to/TO sell/VB [A-10s/NP] abroad/RB ,/, [the/AT Tribune/NP/NP] said/VBD ./.

[The/AT newspaper/NN] said/VBD [one/CD source/NN] reported/VBD that/CS after/CS [Millman/NP/NP] made/VBD [his/PP$ pitch/NN] ,/, [Meese/NP/NP] ordered/VBD [Jenkins/NP/NP] to/TO prepare/VB [a/AT memo/NN] on/IN [behalf/NN] of/IN [Fairchild/NP/NP] ./.

[Memos/NP***] signed/VBD by/IN [Meese/NP/NP] ,/, stressing/VBG [the/AT importance/NN] of/IN [Fairchild/NP/NP 's/$ arranging/VBG sales/NNS] in/IN [Third/NP/NP World/NP/NP countries/NNS] ,/, were/BED sent/VBN to/IN [the/AT State/NP/NP Department/NP/NP] and/CC [the/AT Air/NP/NP Force/NP/NP] ./.

[Millman/NP/NP] did/DOD not/* return/VB [telephone/NN calls/NNS] to/IN [his/PP$ office/NN] and/CC [referral/NN numbers/NNS] [Monday/NR] ,/, [the/AT Tribune/NP/NP] said/VBD ./.