An Alternative Approach to Tagging

Max Silberztein

► To cite this version:

Max Silberztein. An Alternative Approach to Tagging. Proceedings of NLDB 2007. LNCS series, Springer-Verlag Eds, Jan 2007, Paris, France. hal-02435914

HAL Id: hal-02435914

<https://hal.science/hal-02435914>

Submitted on 11 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Alternative Approach to Tagging

Max Silberztein

LASELDI, Université de Franche-Comté.
30 rue Mégevand, 25000 Besançon, France
max.silberztein@univ-fcomte.fr

Abstract. NooJ is a linguistic development environment that allows users to construct large formalised dictionaries and grammars and use these resources to build robust NLP applications. NooJ's approach to the formalisation of natural languages is bottom-up: linguists start by formalising basic phenomena such as spelling and morphology, and then formalise higher and higher linguistic levels, moving up towards the sentence level. NooJ provides parsers that operate in cascade at each individual level of the formalisation: tokenizers, morphological analysers, simple and compound terms indexers, disambiguation tools, syntactic parsers, named entities annotators and semantic analysers. This architecture requires NooJ's parsers to communicate via a Text Annotation Structure that stores both correct results and erroneous hypotheses (to be deleted later).

Keywords: NooJ. Linguistic Development Environment. Robust NLP applications.

1 Introduction

NooJ is a linguistic development environment that allows users to construct large linguistic resources in the form of electronic dictionaries and grammars and to apply these resources to large texts and build various robust NLP applications.¹

NooJ's approach to the formalisation of natural languages is bottom-up: linguists start by formalising basic phenomena, such as spelling, morphology and lexicon, and then use these basic levels of description to formalise higher and higher linguistic levels, moving up towards the syntactic and semantic levels. This bottom-up approach is complemented by an accumulative methodology that allows a community of users to share and re-use individual resources, as well as a number of tools (e.g. concordances, contract enforcers and debuggers) that help users maintain the integrity of large resources.

Parallel to these various levels of formalisation, NooJ provides a number of parsers that operate in cascade at each individual level of the formalisation: at the character level (tokenizer and sentence recogniser), morphology (inflectional and derivational analyser), lexical (simple and compound words recognisers), local syntax

¹ NooJ is freeware. See: <http://www.nooj4nlp.net> to download the software and its documentation.

(disambiguation), structural syntax (frozen and semi-frozen expressions recogniser, syntactic parser) and transformational syntax (semantic analyser).

2 A third type of NLP tool

Most available NLP tools follow one of two different and incompatible approaches.

On the one hand, some linguistic parsers aim at formalising natural languages, usually at the syntactic and semantic levels. Following Chomsky's discussion of the inadequacies of finite-state machines for NLP [1], researchers have invented and refined several computational devices and their corresponding formalisms capable of representing complex, non finite-state syntactic phenomena, such as unification-based parsers that deal with various types of agreement constraints.

Unfortunately most of these parsers, while powerful enough to compute a variety of complex syntactic analyses, are not adapted to the processing of very simple but cost-intensive phenomena, such as locating multi-word expressions in texts by accessing a dictionary of over 200,000 entries, performing morphological analysis of Hungarian texts, etc. Moreover, they are not capable of parsing large corpora in real-time, and therefore cannot be used as online corpus processing tools, nor can they be used as linguistic engines for "basic" applications such as search engines.

On the other hand, some NLP tools aim at facilitating the implementation of NLP applications such as search engines, automatic construction of abstracts, corpus processors, information extraction, etc. These tools often include very efficient parsers based on finite-state technology, and can indeed be used to parse large quantities of texts. Unfortunately, these tools include at one point or another several algorithms that make them unsuitable to the formalisation of natural languages, such as a statistical tagger that aims at producing "reasonably good" results -- which is to say a number of incorrect ones -- as well as heuristics to get rid of ambiguities-- even when sentences are genuinely ambiguous-- etc.

NooJ shares with the above-mentioned linguistic tools the goal of providing linguists with a way to formalise natural languages precisely, and at the same time includes several efficient finite-state tools to parse large texts and process large linguistic resources. This can be done because in NooJ's bottom-up architecture, each level of analysis is processed by a different, specialised (and therefore efficient) computational device. In other words, instead of using one single powerful (and inefficient) computational device to process all kinds of linguistic phenomena, we assume that natural languages are sets of very different phenomena, each of them requiring a specialized mechanism and associated parser; in particular, simple devices such as finite-state machines, which constitute very natural tools to represent a large number of linguistic phenomena, should not be thrown away because they are not adequate to represent other, sometimes even exotic, phenomena.

3 Atomic Linguistic Units

In NooJ, the term **Atomic Linguistic Units** (ALUs) refers to the smallest elements of a given language that are associated with linguistic information. By definition, these ALUs constitute the vocabulary of the language. They can and must be systematically described in extension, because some of, or all their properties cannot be computed from their components.

NooJ's first level of text analysis is at the character level. Characters are classified as **letters** and **delimiters**. **Tokens** are sequences of letters between delimiters. Based on these three definitions, NooJ distinguishes four types of ALUs:

- **Simple Word**: any ALU spelled as a token, e.g. *table*
- **Affix**: any ALU spelled as a subsequence of letters in a token, e.g. *re-*, *-able*
- **Multi-Word Unit** (MWU): any ALU spelled as a sequence of letters and delimiters, e.g. *as a matter of fact* (the space character is not a letter, hence it is a delimiter)
- **Frozen Expression**: any MWU that accepts insertions, e.g. *take ... into account*

This classification is adequate for any written language, although some languages (e.g. English) require the description of very few affixes whereas others (e.g. Hungarian) require a large morphological system. The vocabulary of Romance languages probably contains five times more MWUs than simple words, whereas Germanic languages have very few MWUs because these units are not spelled with spaces, and hence are processed by NooJ as simple words.²

Obviously, it is important for any NLP application to be able to process any type of ALUs, including MWUs. Questions about MWUs -- their definition, how to automatically recognise them in texts, how to process them, etc. -- have initiated much interest in computational linguistics. Unfortunately, there seems to be confusion about what people mean by MWUs (or compounds, or complex terms, collocations, etc.). For instance, [2] studies "transparent" noun compounds and discusses methods for analysing them. In the NooJ framework, this is contradictory: either these objects are truly MWUs (i.e. **Atomic Linguistic Units**) that must be listed and explicitly described in dictionaries (and therefore don't need to be analysed), or they are transparent and can be analysed automatically (and therefore do not need to be listed explicitly in a dictionary). Note finally that more precise analyses (such as [3]) show that so-called "transparent" MWUs have in fact some degrees of "opacity" that would require robust NLP application to list and describe them explicitly in a dictionary.

Statisticians often equate compounds with collocations, i.e. sequences of tokens that occur together frequently. But trying to characterise MWUs by computing the co-occurrence of their components negates their atomicity.³ Moreover, MWUs (just like

² On the other hand, Germanic analysable tokens, such as "*Schiffahrtsgesellschaft*" must be processed as sequences of affixes: "*Schiff fahrt s gesellschaft*".

³ In the same manner that one should not try to prove the fact that the token "apartment" is an English word from the fact that "apart" and "ment" often occurs together. In fact, tokenizers used by most taggers and statistical parsers naively use the blank character to characterise linguistic units. In NooJ, MWUs are ALUs just like simple words; the fact that they include

simple words) can be either frequent or rare in texts, and MWUs with a low frequency (say, less than 3 occurrences in a large corpus), are typically left out by statistical methods.

4 Processing Multi-Word Units

NLP applications would gain significant advantages were they to recognise MWUs in texts just by consulting pre-built dictionaries, or by accessing tagged texts in which these units were fully represented. These include:

(1) Applications that extract and count words in order to characterise parts of a corpus for information retrieval, summarisation, question answering, etc. For instance, indexing tokens such as “plant”, “control”, “group”, “board”, etc. produces too many useless hits, whereas indexing MWUs such as “wild plant” or “power plant” produces much more precise results. Moreover, distinguishing terms (e.g. *parking lots*) from non-terms (ex. ... *lots of cars*...) will no longer raise complicated issues.

(2) Ranking systems subsequently select among the texts retrieved by the search engine those that will be most useful to the user. Ranking programs are said to be based on *word* counts, although what they really count are *tokens*. One problem is that most terms have abbreviated variants that are often more frequent than their explicit form. For instance, in articles that include the term *school board*, the abbreviated variant “board” will often occur more frequently than the explicit form. Current ranking systems fail to recognise a term when only its abbreviation occurs. The following enhancement could easily be implemented: when a MWU is recognised, all its variants that occur within a certain context should be linked to it. For instance, if an indexer found the term “school board” in a text, the subsequent utterances of the token “board” within the same paragraph could be linked to it. Thus, the term would receive a much higher frequency even though its explicit form might have occurred only once.

(3) In Question Answering applications, one might want to retrieve a text such as:

... Last presidential election was won by John Smith ...

from queries such as: ‘What was the election about?’ or ‘Who was elected?’. This is typically done by using a rule such as “WORD election = people elect WORD”. But then, we need to inhibit this rule for a number of terms such as:

local election, runoff election, general election, free election...

blanks or other delimiters is not relevant to their status, and does not even complicate their automatic recognition in texts.

because people do not elect “locals”, “runoffs”, “generals” or “freedom.” The only proper way⁴ to correctly analyse relevant phrases while rejecting all irrelevant ones is to enter explicitly the rules that are associated with each MWU:

presidential election = people elect someone president
party election = a party elects its leader/representative
free election = an election is free = people are free to participate in the election

(4) In multilingual applications, such as the teaching of a second language or Machine Translation (MT), text mining of multilingual corpora, etc. it is crucial to process MWUs. For instance, if the adverb “as a matter of fact” is treated as a sequence of five words, a translator will produce the incorrect French text “comme une matière de fait” rather than the correct term “à vrai-dire.” Reciprocally, the French MWU “traitement de texte” must be translated as “word processor”, rather than as “treatment of text” or “text treatment.”

MT is one of the most ambitious NLP applications because it requires a precise description of both source and target languages. Moreover, as opposed to syntactic parsers that produce abstract data structures or information retrieval systems that access texts containing billions of words, any native speaker can easily judge the quality of its results. Appendix 1 contains an extract from a French newspaper along with a translation that was produced by current automatic machine translation systems: the reader can judge the quality (or lack thereof) of the result, and evaluate the enormous advantage of having the computer access a dictionary of MWUs.

5 Tagging All ALUs

Tagged texts are mainly used as inputs for syntactic or statistical parsers. Unfortunately, most taggers associate tags to simple words without taking other types of ALUs into account⁵ Let us illustrate this by studying the following text sample:⁶

Battle-tested Japanese industrial managers here always
buck up nervous newcomers with the tale of **the first of**
their countrymen to visit Mexico, a boatload of samurai
warriors blown ashore 375 years ago. From the beginning,
it took a man with extraordinary qualities to succeed in
Mexico, says Kimihide Takimura, president of Mitsui
group's Kensetsu Engineering Inc. unit.

Fig. 1. Multi-word units are underlined in a text.

⁴ In an equivalent manner, [4] discusses how to link MWUs to their semantic representation, e.g. *prefabricated housing* and *produce (factory, house)*.

⁵ Affixes, MWUs and frozen expressions are usually not taken into account by taggers, even when the goal is to locate Noun Phrases, see for instance the NP recogniser described in [5].

⁶ The tagging of this text is commented on in [6], but there is no mention of its MWUs.

-- “Battle-tested”: metaphorical adjective synonymous with “experienced.” One cannot expect a semantic parser to infer the idiosyncratic analysis, or an automatic translator to compute the correct word (e.g. the French adjective “aguerris”) from the two components “Battle” (“Bataille”) and “tested” (“testés”).

-- “industrial manager”: must *not* be analysed by the semantic rule:

Industrial N = N is made/created by industrial methods

although that this very productive rule does apply to numerous sequences, such as “industrial diamond”, “industrial food”, “industrial soap”, “industrial chicken”, etc.

-- “to buck up”: not to be processed as the free verb *to buck* that would be followed by the locative preposition “up”;

-- “a boatload of”: must be tagged as a determiner, and *not* as a noun, because we want the following human noun *warriors* to be the head of the noun phrase;

-- “samurai warriors” = explicit variant of the term “samurais”; not to be analysed as “N warriors” such as “Japanese warriors” or “monk warriors” (all samurais *are* warriors, by definition);

-- “blown ashore”: not to be processed as the free verb *to blow* followed by the locative adverb “ashore”, even though the metaphor is clearer (to a human reader, *not* to an automatic parser)

-- “from the beginning”: adverbial complement of date

-- “It takes NP to VP”: frozen expression synonymous with “Only NP can VP”

Ignoring any of these MWUs during the tagging process is extremely costly, because it leads to unavoidable mistakes during the subsequent parsing. Note that none of these ALUs are indeed correctly translated by the machine translation systems mentioned in Appendix 1.

A more useful tagger, i.e. one that takes all ALUs (including MWUs and frozen expressions) into account would produce a result similar to the following one:⁷

```
<Battle-tested,A> <Japanese,A> <industrial
manager,N+Hum+p> <here,ADV+Loc> <always,ADV> <buck
up,V+PR> <nervous,A> <newcomer,N+Hum+p> <with,PREP>
<the,DET+s> <tale,N+s> <of,PREP> <the first of
their,DET+p> <countryman,N+Hum+m+p> <to,PREP>
<visit,V+INF> <Mexico,N+Proper+Loc>, <a boatload
of,DET+p> <samurai warrior,N+Hum+p> <blow ashore,V+PP>
<375 years ago,ADV+Date>.
<From the beginning,ADV+Date>, <it,PRO+EXP01+3+s>
<take,V+EXP01+PT> <a,DET+s> <man,N+Hum+m+s> <with,PREP>
<extraordinary,A> <quality,N+p> <to,PREP+EXP01>
<succeed,V+INF> <in,PREP> <Mexico,N+Proper+Loc>,
<say,V+PR+3+s> <Kimihide Takimura,N+Proper+Hum+s>,
```

⁷ In NooJ, tags (between brackets) associate each ALU with its lemma and linguistic information. “A” stands for Adjective, “N” for Noun, “V” for Verb. Morphological, syntactic and distributional features are prefixed by a character “+”, e.g. “+Hum” stands for human; “+t” stands for transitive; +p stands for plural. See [7] for a description of NooJ’s codes.

<president,N+Hum+s> <of,PREP> <Mitsui,N+Proper+Company>
 <group,N+s> <'s,POSS> <Kensetsu Engineering
 Inc.,N+Proper+Company> <unit,N+s>.

Fig. 2. A Part of Speech tagger that tags all types of ALUs.

In this tagged text, all ALUs are represented by one tag, exactly like simple words. The constituents of the frozen expression “it takes NP to VP” are marked with the I.D. number “EXP01” in order to allow subsequent parsers to retrieve the properties of the expressions and link the components together.⁸

In this sample of 59 tokens, 31 tokens are in fact components of MWUs, frozen expressions and semi-frozen expressions (such as “375 years ago”), and thus are usually tagged with misleading tags (such as “boatload=N”), incorrect ones (e.g. “ago=ADV”), redundant ones (“warriors”) or irrelevant ones (“Engineering”) by traditional taggers. The high frequency of MWUs and frozen expressions is far from being exceptional, as [9] showed. Another example is displayed in Appendix 2.

6 An Ambiguous Text Annotation Structure

The best possible outcome would be to produce a tagged text such as the one in figure 2; unfortunately it is not generally possible to remove all ambiguities without performing a complex series of analyses. For instance, in the following context:

... There is a round table in this building ...

there *are* two possible analyses: either the MWU “round table” (= a meeting) occurs, or the sequence of two simple words “round” and “table” (= a table with a round shape). It would require a complex discourse analysis of a broader context of this sentence to solve this ambiguity, and any local parser, such as a statistical-based algorithm trained on some other text, or a simple program that either ignores MWUs, or always tags them, is going to produce a significant number of errors. In NooJ’s architecture however, no correct hypothesis can be missing (a “no silence” constraint), because in NooJ’s entire flow of analyses, later parsers cannot go back to correct their inputs. In other words, if it is possible that the MWU “round table” occurs in the text, it has to be tagged, and, at the same time, if it is possible that the two ALUs “round” and “table” occur in the text, they too have to be tagged. Generally, ambiguities can be solved only at a phase well after the tagging process.

Although annotations or XML tags are widely used to represent linguistic units such as named entities, very few taggers have used them to represent the massively ambiguous results of the lexical analysis of a text.⁹

In order to maintain both hypotheses alive during all the analyses, NooJ’s consecutive parsers communicate their results via a Text Annotation Structure which

⁸ I assume that there is some provision to process idiomatic and frozen expressions at the syntactic analysis stage.

⁹ INTEX used finite-state transducers to represent the results of its lexical parser, see [9].

stores both correct results and erroneous hypotheses. Each of NooJ's parsers -- at the character, morphological, lexical, syntactic and semantic levels -- takes a Text Annotation Structure as its input, and produces a Text Annotation Structure, such as the one shown in Appendix 3.

NooJ's annotation system represents all ALUs in the same exact way, whether they were produced by a morphological parser (affixes), by a lookup of a dictionary (simple words and MWUs) or by local grammars (for discontinuous frozen expressions). Indeed, after NooJ has completed the full lexical analysis of a text, the following syntactic parsers do not even need to know the original type of each ALU that they are processing.

Processing Text Annotation Structures is more complex than processing simple sequences of tags. For instance, processing a simple query such as locating in a text all the word forms "the" followed by a plural Noun, involves using a parser that can read both the text and its (ambiguous) annotations; that is substantially heavier than running a simple "grep" command on traditionally tagged texts.

An important advantage of this architecture is that several applications, such as Named Entity Recognisers, can run on Text Annotation Structures even if these have not been fully disambiguated. For instance, to recognise the sequence "on the 12th of May, 2006", NooJ's TIMEX recogniser does not need to disambiguate the word form "May" (the verb or the month name): indeed one can argue that the very fact that TIMEX has annotated the previous sequence as an adverb makes the disambiguation of the word form "May" trivial, and even unnecessary.

7 Conclusion

NooJ is currently used as a linguistic tool to formalise more than 20 languages, and NooJ modules are already available for download for Arabic, Armenian, Catalan, Chinese, English, French, Hebrew, Hungarian, Italian, Latin, Portuguese and Spanish. These modules take into account the four types of NooJ ALUs: affixes, simple words, MWUs and frozen expressions.

Because NooJ's architecture guarantees a "zero silence" (though at the cost of some inevitable noise), it becomes possible to build truly robust NLP applications. Several NLP applications already use NooJ as their linguistic engine, including the current French project VODEL from the National Agency for Research (ANR), which aims at parsing automatically medical texts written in French.

References

1. Chomsky, Noam. *Syntactic Structures*. La Haye: Mouton, 1957.
2. Barker, Ken; Szpakowicz, Stan. Semi-Automatic Recognition of Noun Modifier Relationships. In *Proceedings of COLING-ACL'98*. Montréal: 1998.
3. Silberstein, Max. Les groupes nominaux productifs et les noms composés lexicalisés. In *Linguisticae Investigationes* #17:2. John Benjamins: Amsterdam, 1993.

4. Lewis, David; Sparck Jones, Karen. Natural Language Processing for Information Retrieval. In *Communications of the ACM* #39:1. ACM: New York, 1996.
5. Church, Ken. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of Second Conference on Applied Natural Language Processing (ANLP'88)*: Austin TX, 1988.
6. Marcus, Mitchell; Santorini, Beatrice; Marcinkiewicz, Mary Ann. Building a large annotated corpus of English: the Penn Treebank. In *Computational Linguistics* #19, The MIT Press, 1993.
7. Silberztein, Max. NooJ's Manual can be downloaded from <http://www.nooj4nlp.net>, 2002.
8. Silberztein, Max. *Dictionnaires électroniques et analyse automatique de textes*. Masson. Paris: 1993.
9. Silberztein, Max. Dictionnaires électroniques et comptage des mots. In *Proceedings of Troisièmes Journées d'Analyse des Données Textuelles (JADT)*. Rome: 1995.

Appendix 1: Translation of a sample from *Le Monde*, January 1994

In the following French text, I have underlined MWUs (e.g. vie privée) and verbal expressions (tomber sous le coup de la loi). Semi-frozen expressions are in bold.

"Bonne année". **L'an 1994** aura commencé **samedi à zéro heure**, qui en douterait ? Et pourtant, cette évidence est pour le moins trompeuse. C'est ainsi que, décalage horaire aidant, les Australiens de Sydney fêtent la Saint-Sylvestre **avec neuf heures d'avance** sur nous et les Américains de Los Angeles **avec neuf heures de retard**. Un détail ? Admettons. Mais que dire, alors, des musulmans pratiquants qui, eux, se considèrent à la mi-1414, et célébreront leur "nouvel an" (l'hégire) **le 10 juin** ? Quant aux juifs, leur **année 5754** a débuté **le 16 septembre**. Incontestablement, le temps est une notion toute relative. Les physiciens savent cela depuis longtemps. Les politiques et les religieux aussi. Mais, grands amateurs de certitudes, ces derniers se sont alliés aux scientifiques pour tenter d'offrir aux hommes un guide chronologique susceptible de régler une vie sociale qu'ils voulaient **aussi harmonieuse que possible**. Ce n'est pas un hasard si le calendrier grégorien que nous utilisons actuellement **en Occident** fut imposé (sous peine d'excommunication !) **il y a quatre siècles** par un pape chagriné de voir le jour de Pâques s'éloigner de la période que lui avaient fixée ses prédécesseurs **en l'an 325** !

Following is the best result produced by the French-to-English translators available at: amikai.com, freetranslation.com, officeupdate.lhsl.com, reverso.net, systansoft.com, t-mail.com, tranexp.com, translationwave.com. Mistakes are crossed out:

" ~~Good year~~ ". ~~Will the year 1994~~ have begun Saturday at zero hour, ~~which would doubt it?~~ And yet, this obviousness is

at the very least misleading. Thus, ~~time shift~~ helping, the Australian ~~ones~~ of Sydney celebrate New Year's Eve ~~with nine hours in advance on us and the Americans of Los Angeles with nine hours of delay~~. A detail? Let us admit [it]. But what to say, then, of the ~~Moslems practise~~ who, ~~them~~, are considered with ~~semi-1414~~, and will celebrate their " new year " (the *hégire*) on June 10? As for the Jews, their year 5754 began on September 16. Incontestably, time is a quite relative concept. The physicists [have known] that for a long time. ~~Policies and monks~~ too. But, ~~large amateurs~~ of certainty, the latter ~~were combined to~~ the scientists to try to offer to ~~the men~~ a chronological guide suitable ~~for~~ regulate a ~~social life~~ which they wanted ~~possible as harmonious as~~. ~~It is not a chance if~~ the Gregorian calendar that we currently use ~~in Occident were~~ imposed (under penalty of excommunication!) four centuries ago by a pope ~~grained~~ to see ~~the Easter Day~~ moving away ineluctably from the period that its predecessors ~~in year 325~~ had fixed ~~to him~~!

Note that 18 of the mistakes would have been avoided by a simple lookup of a bilingual dictionary that would include the following MWUs:

Bonne année = Happy new year; qui en douterait ? = who would doubt it?; et pourtant = and yet; pour le moins = at the very least; C'est ainsi que = thus; décalage horaire = time difference; la Saint-Sylvestre = New Year's Eve; musulman pratiquant = practising Muslim; se considérer = to consider oneself to be; nouvel an = new year; depuis longtemps (+Présent) = for a long time (+Present Perfect); grands amateurs = big fans; s'allier à = to ally oneself with; vie sociale = life in society; Ce n'est pas un hasard si = it is not by accident if; calendrier grégorien = Gregorian calendar; sous peine de = under penalty of; jour de Pâques = Easter

Appendix 2: Sample from the Wall Street Journal, January 1996

I have underlined various types of MWUs: nouns (e.g. monetary union), adverbs (in the process), proper names (U.S.), prepositions (according to) and verbal expressions (cast a shadow on). Auxiliary and transparent verbs have been crossed out (e.g. ~~may be~~-heading). In bold are semi-frozen expressions for determiners (**one of the best**), adverbs of date (**in 1995**), and of duration (**in the space of one year**).¹⁰

Europe ~~may be~~-heading toward economic convergence and monetary union, but its stock markets still set differing courses. Just as **in 1995**, when some countries produced high double-digit returns while others posted substantial losses, investors ~~can~~-expect a grab bag of returns this year. On average European stocks rose 12.3% **last year**, according to the Eurotop 100 index, which is far below the stratospheric returns of U.S. stocks. But returns in individual European countries ranged **as high as** 20.3% in Zurich, 17.1% in Dublin and 16.9% in London and **as low as** a decline of 8.36% in Helsinki, 7.81% in Milan and 0.49% in Paris. And the performance of specific sectors also varied to a large degree with financial, technology and pharmaceutical stocks producing returns ranging from 21% to 50%. Paper, steel and heavy machinery issues had a rough year, however, with losses ranging from 4% to 26%.

LONDON: Merger activity and low interest rates sent U.K. stocks soaring **last year**. The Financial Times-Stock Exchange 100 Share Index climbed 16.9%, finishing the year at 3689.3. The U.K. market had **one of the best** performances in Europe, with the index gradually rising from a low of 2876.60 **at the beginning of the year**.

A good year for British government bonds, or gilts, also ~~helped~~ propel U.K. stocks upward. The average yield dropped to 7.4% from 8.7% **at the beginning of the year**. Richard Kersley, U.K. market strategist with BZW Ltd. in London, says the change from rising to falling interest rates sparked **much of the stock market climb in 1995**. **In mid-December**, the British government cut the base interest rate by a quarter percentage point to 6.50% and analysts say further interest rate cuts ~~are~~-expected **early this year**. FRANKFURT: The strengthening of the mark cast a shadow on Germany's stock prices **throughout last year**, forcing earnings revisions downward **for 1995, 1996 and 1997** in export-oriented companies. The DAX Index, the key gauge of 30 stocks traded at the Frankfurt Stock Exchange, fell to 2253.88 points, chalking up only a 6.54% gain **since the beginning of the year**. **Last year**, it fell to a low of 1893.63 and climbed to a high of 2320.22. But on the positive side, declining German bond yields helped the stock market. **In the space of one year**, the **10-year** benchmark bond yield ~~has~~-fallen from 7.17% to its current 6.02%.

¹⁰ Numerical expressions ("20.3%", "2253.88 points") and intervals ("from 21% to 50%") could also be recognised by a local grammar, and processed as Atomic Linguistic Units.

Appendix 3: A text annotation structure

Right after the lookup of NooJ's dictionaries and morphological grammars, the text annotation structure stores all lexical hypotheses and the corresponding ambiguities, such as the ones between “round” followed by “table” and “round table”. Discontinuous linguistic units, such as “to take ... into account”, are also represented.

The screenshot shows the NooJ Community Edition interface with the file "_take into account.not" open. The main window displays a text annotation structure for the sentence "He took the round table into account". The interface includes a menu bar (File, Edit, Lab, Project, Windows, Info, TEXT), a toolbar, and a status bar.

The text annotation structure is displayed in a table-like format with columns for tokens and their corresponding linguistic units. The tokens are: "he", "PRO", "take", "into", "account", "V+PT+1+2+3+s+tp", "take", "V+PT+1+2+3+s+tp", "the", "DET", "round", "table", "N+XN+Conc+1+s", "round", "PREP", "round", "V+INF", "round", "V+PR+1+2+s", "round", "V+PR+1+2+3+tp", "round", "A+N", "round", "ADV", "round", "N+s".

The linguistic units are listed in the right column, with arrows indicating the mapping from tokens to units. For example, "he" maps to "he, PRO", "take" maps to "take, V+PT+1+2+3+s+tp", "into" maps to "into, PREP", "account" maps to "account, N+s", and "round" maps to "round, A+N".

The interface also includes a "Show Text Annotation Structure" checkbox and a "Cancel" button.