Generic NLP Technologies: Language, Knowledge and Information Extraction

Junichi Tsujii
Department of Information Science, Faculty of Science
University of Tokyo, JAPAN
And
Centre for Computational Linguistics, UMIST, UK

1 Introduction

We have witnessed significant progress in NLP applications such as information extraction (IE), summarization, machine translation, cross-lingual information retrieval (CLIR), etc. The progress will be accelerated by advances in speech technology, which not only enables us to interact with systems via speech but also to store and retrieve texts input via speech.

The progress of NLP applications in this decade has been mainly accomplished by the rapid development of corpus-based and statistical techniques, while rather simple techniques have been used as far as the structural aspects of language are concerned.

In this paper, we will discuss how we can combine more sophisticated, linguistically elaborate techniques with the current statistical techniques and what kinds of improvement we can expect from such an integration of different knowledge types and methods.

2 Argument against linguistically elaborate techniques

Throughout the 80s, research based on linguistics had flourished even in application oriented NLP research such as machine translation. Eurotra, a European MT project, had attracted a large number of theoretical linguists into MT and the linguists developed clean and linguistically elaborate frameworks such as CTA-2, Simple Transfer, Eurotra-6, etc.

ATR, a Japanese research institute for telephone dialogue translation supported by a consortium of private companies and the Ministry of Post and Communication, also adopted a linguistics-based framework, although they changed their direction in the later stage of the project. They also adopted sophisticated plan-based dialogue models as well at the initial stage of the project.

However, the trend changed rather drastically in the early 90s and most research groups with practical applications in mind gave up such strategies and switched to more corpus-oriented and statistical methods. Instead of sentential parsing based on linguistically well founded grammar, for example, they started to use simpler but more robust techniques based on finite-state models. Neither did knowledge-based techniques like plan-recognition, etc. survive, which presume explicit representation of domain knowledge.

One of the major reasons for the failure of these techniques is that, while these techniques alone cannot solve the whole range of problems that NLP application encounters, both linguists and AI researchers made strong claims that their techniques would be able to solve most, if not all, of the problems. Although formalisms based on linguistic theories can certainly contribute to the development of clean and modular frameworks for NLP, it is rather obvious that linguistics theories alone cannot solve most of NLP's problems. Most of MT's problems, for example, are related with semantics or interpretation of language which linguistic theories of syntax can hardly offer solutions for (Tsujii 1995).

However, this does not imply, either, that frameworks based on linguistic theories are of no use for MT or NLP application in general. This only implies that we need techniques complementary to those based on linguistic

theories and that frameworks based on linguistic theories should be augmented or combined with other techniques. Since techniques from complementary fields such as statistical or corpus-based ones have made significant progresses, it is our contention in this paper that we should start to think seriously about combining the fruits of the research results of the 80s with those of the 90s.

The other claims against linguistics-based and knowledge-based techniques which have often been made by practical-minded people are :

(1) Efficiency: The techniques such as sentential parsing and knowledge-based inference, etc. are slow and require a large amount of memory

(2) Ambiguity of Parsing: Sentential parsing tends to generate thousands of parse results from which systems cannot choose the correct one.

(3) Incompleteness of Knowledge and Robustness: In practice one cannot provide systems with complete knowledge. Defects in knowledge often cause failures in processing, which result in the fragile behavior of systems.

While these claims may have been the case during the 80s, the steady progress of such technologies have largely removed these difficulties. Instead, the disadvantages of current technologies based on finite state technologies, etc. have increasingly become clearer; the disadvantages such ad-hocness and opaqueness of systems which prevent them from being transferred from an application in one domain to another domain.

3 The current state of the JSPS project

In a five-year project funded by JSPS (Japan Society of Promotion of Science) which started in September 1996, we have focussed our research on generic techniques that will be used for different kinds of NLP application and domains.

The project comprises three university groups from the University of Tokyo, Tokyo Institute of Technology (Prof. Tokunaga) and Kyoto University (Dr. Kurohashi), and coordinated by myself (at the University of Tokyo). The University of Tokyo has been engaged in development of software infrastructure for efficient NLP, parsing technology and ontology building from texts, while the groups of Tokyo Institute of Technology and Kyoto University have been responsible for NLP application to IR and Knowledge-based NLP techniques, respectively.

Since we have delivered promising results in research on generic NLP methods, we are now engaged in developing several application systems that integrate various research results to show their feasibility in actual application environments. One such application is a system that helps biochemists working in the field of genome research.

The system integrates various research results of our project such as new techniques for query expansion and intelligent indexing in IR, etc. The two results to be integrated into the system that we focus on in this paper are IE using a full-parser (sentential parser based on grammar) and ontology building from texts.

IE is very much in demand in genome research, since quite a large portion of research is now being targeted to construct systems that model complete sequences of interaction of various materials in biological organisms. These systems require extraction of relevant information from texts and its integration in fixed formats. This entails that the researchers there should have a model of interaction among materials, into which actual pieces of information extracted from texts are fitted. Such a model should have a set of classes of interaction (event classes) and a set of classes of entities that participate in events. That is, the ontology of the domain should exist. However, since the building of an ultimate ontology is, in a sense, the goal of science, the explicit ontology exists only in a very restricted and partial form. In other words, IE and Ontology building are inevitably intertwined here.

In short, we found that IE and Ontology

building from texts in genome research provide an ideal test bed for our generic NLP techniques, namely software infrastructure for efficient NLP, parsing technology, and ontology building from texts with initial partial knowledge of the domain.

4 Software Infrastructure and Parsing Technology

While tree structures are a versatile scheme for linguistic representation, invention of feature structures that allow complex features and reentrancy (structure sharing) makes linguistic representation concise and allows declarative specifications of mutual relationships among representation of different linguistic levels (e.g.: morphology, syntax, semantics, discourse, etc.). More importantly, using bundles of features instead of simple non-terminal symbols to characterize linguistic objects allow us to use much richer statistical means such as ME (maximum entropy model), etc. instead of simple probabilistic CFG. However, the potential has hardly been pursued yet mostly due to the inefficiency and fragility of parsing based on feature-based formalisms.

In order to remove the efficiency obstacle, we have in the first two years devoted ourselves to the development of :

(A) Software infrastructure that makes processing of feature-based formalisms efficient enough both for practical application and for combining it with statistical means.

(B) Grammar (Japanese and English) with wide coverage for processing real world texts (not examples in textbooks of linguistics). At the same time, processing techniques that make a system robust enough for application.

(C) Efficient parsing algorithm for linguistics-based frameworks, in particular HPSG.

We describe the current states of these three in the following.

(A) Software Infrastructure (Miyao 2000):

We designed and develop a programming system, LiLFeS, which is an extension of Prolog for expressing typed feature structures instead of first order terms. The system's core engine is an abstract machine that can process features and execute definite clause program. While similar attempts treat feature structure processing separately from that of definite clause programs, the LiLFeS abstract machine increases processing speed by seamlessly processing feature structures and definite clause programs.

Diverse systems, such as large scale English and Japanese grammar, a statistical disambiguation module for the Japanese parser, a robust parser for English, etc., have already been developed in the LiLFeS system.

We compared the performance of the system with other systems, in particular with LKB developed by CSLI, Stanford University, by using the same grammar (LinGo also provided by Stanford University). A parsing system in the LiLFeS system, which adopts a naive CKY algorithm without any sophistication, shows similar performance as that of LKB which uses a more refined algorithm to filter out unnecessary unification. The detailed examination reveals that feature unification of the LiLFeS system is about four times faster than LKB.

Furthermore, since LiLFeS has quite a few built-in functions that facilitate fast subsumption checking, efficient memory management, etc., the performance comparison reveals that more advanced parsing algorithms like the one we developed in (C) can benefit from the LiLFeS system. We have almost finished the second version of the LiLFeS system that uses a more fine-grained instruction set, directly translatable to naive machine code of a Pentium CPU. The new version shows more than twice improvement in execution speed, which means the naive CKY algorithm without any sophistication in the LiLFeS system will outperform LKB.

(B) Grammar with wide coverage (Tateisi 1998; Mitsuishi 1998):

While LinGo that we used for comparison is an interesting grammar from the view point of linguistics, the coverage of the grammar is rather restricted. We have cooperated with the University of Pennsylvania to develop a grammar with wide coverage. In this cooperation, we translated an existing wide-coverage grammar of XTAG to the framework of HPSG, since our parsing algorithms in (C) all assume that the grammar are HPSG. As we discuss in the following section, we will use this translated grammar as the core grammar for information extraction from texts in genome science.

As for wide-coverage Japanese Grammar, we have developed our own grammar (SLUNG) . SLUNG exploits the property of HPSG that allows under-specified constraints. That is, in order to obtain wide-coverage from the very beginning of grammar development, we only give loose constraints to individual words that may over-generate wrong interpretations but nonetheless guarantee correct ones to be always generated.

Instead of rather rigid and strict constraints, we prepare 76 templates for lexical entries that specify behaviors of words belonging to these 76 classes. The approach is against the spirit of HPSG or lexicalized grammar that emphasizes constraints specific to individual lexical items. However, our goal is first to develop wide-coverage grammar that can be improved by adding lexical-item specific constraints in the later stage of grammar development. The strategy has proved to be effective and the current grammar can produce successful parse results for 98.3 % of sentences in the EDR corpus with high efficiency (0.38 sec per sentence for the EDR corpus). Since the grammar overgenerates, we have to choose single parse results among a combinatorially large number of possible parses. However, an experiment shows that a statistic method using ME (we use the program for ME developed by NYU) can select around 88.6 % of correct analysis in terms of dependency relationships among ! ! bunsetsu's - the phrases in Japanese).

(C) Efficient parsing algorithm (Torisawa 2000):

While feature structure representation provides an effective means of representing linguistic objects and constraints on them, checking satisfiability of constraints by linguistic objects, i.e. unification, is computationally expensive in terms of time and space. One way of improving the efficiency is to avoid unification operations as much as possible, while the other way is to provide efficient software infrastructure such as in (A). Once we choose a specific task like parsing, generation, etc., we can devise efficient algorithms for avoiding unification.

LKB accomplishes such reduction by inspecting dependencies among features, while the algorithm we chose is to reduce necessary unification by compiling given HPSG grammar into CFG. The CFG skeleton of given HPSG, which is semi-automatically extracted from the original HPSG, is applied to produce possible candidates of parse trees in the first phase. The skeletal parsing based on extracted CFG filters out the local constituent structures which do not contribute to any parse covering the whole sentence. Since a large proportion of local constituent structures do not actually contribute to the whole parse, this first CFG phase helps the second phase to avoid most of the globally meaningless unification. The efficiency gain by this compilation technique depends on the nature of the original grammar to be compiled. While the efficiency gain for SLUNG is just two times, the gain for XHPSG (HPSG grammar obtained by translating the XTAG grammar into HPSG) is around 47 times for the ATIS corpus (Tateisi 1998).

5 Information extraction by sentential parsing

The basic arguments against use of sentential parsing in practical application such as IE are the inefficiency in terms of time and space, the fragility of systems based on linguistically rigid frameworks and highly ambiguous parse

results that we often have as results of parsing.

On the other hand, there are arguments for sentential parsing or the deep analysis approach. One argument is that an approach based on linguistically sound frameworks makes systems transparent and easy to re-use. The other is the limit on the quality that is achievable by the pattern matching approach. While a higher recall rate of IE requires a large amount of patterns to cover diverse surface realization of the same information, we have to widen linguistic contexts to improve the precision by preventing extraction of false information. A pattern-based system may end up with a set of patterns whose complex mutual nullify the initial appeal of simplicity of the pattern-based approach.

As we see in the previous section, the efficiency problem becomes less problematic by utilizing the current parsing technology. It is still a problem when we apply the deep analysis to texts in the field of genome science, which tend to have much longer sentences than in the ATIS corpus. However, as in the pattern-based approach, we can reduce the complexity of problems by combining different techniques.

In a preliminary experiment, we first use a shallow parser (ENGCG) to reduce part-of-speech ambiguities before sentential parsing. Unlike statistic POS taggers, the constraint grammar adopted by ENGCG preserves all possible POS interpretations just by dropping interpretations that are impossible in given local contexts. Therefore, the use of ENGCG does not affect the soundness and completeness of the whole system, while it reduces significantly the local ambiguities that do not contribute to the whole parse.

The experiment shows that ENGCG prevents 60 % of edges produced by a parser Based on naive CKY algorithm, when it is applied to 180 sentences randomly chosen from MEDLINE abstracts (Yakushiji 2000). As a result, the parsing by XHPSG becomes four times faster from 20.0 seconds to 4.8 second per sentence, which is further improved by using chunking based on the output of a Named Entity recognition tool to 2.57 second per sentence. Since the experiment was conducted with a naive parser based on CYK and the old version of LiLFeS, the performance can be improved further.

The problems of fragility and ambiguity still remain. XHPSG fails to produce parses for about half of the sentences that cover the whole. However, in application such as IE, a system needs not have parses covering the whole sentence. If the part in which the relevant pieces of information appear can be parsed, the system can extract them. This is one of the major reasons why pattern-based systems can work in a robust manner. The same idea can be used in IE based on sentential parser. That is, techniques that can extract information from partial parse results will make the system robust.

The problem of ambiguity can be treated in a similar manner. In a pattern-based system, the system extracts information when parts of the text match with a pattern, independently of whether other interpretations that compete with the interpretation intended by the pattern exist or not. In this way, a pattern-based system treats ambiguity implicitly. In case of the approach based on sentential parsing, we treat the ambiguity problem by preference. That is, an interpretation that indicates relevant pieces of information exist is preferred to other interpretations.

Although the methods illustrated in the above make IE based on sentential parsing similar to the pattern-based approach, the approach retains the advantages over the pattern-based one. For example, it can prevent false extraction if the pattern that dictates extraction contradicts with wider linguistic structures or with the more preferred interpretations. It keeps separate the general linguistic knowledge embodied in the form of XHPSG grammar that can be used in any domain. The mapping between syntactic structures to predicate structures can also be systematic.

6 Information extraction of named entities using a hidden Markov model

The named entity tool mentioned above, called NEHMM (Collier 2000), has been developed as a generalizable supervised learning method for identifying and classifying terms given a training corpus of SGML marked-up texts. HMMs themselves belong to a class of learning algorithms that can be considered to be stochastic finite state machines. They have enjoyed success in a wide number of fields including speech recognition and part of speech tagging. We therefore consider their extension to the named entity task, which is essentially a kind of semantic tagging of words based on their class, to be quite natural.

NEHMM itself strives to be highly generalizable to terms in different domains and the initial version uses bigrams based on lexical and character features with one state per name class. Data-sparseness is overcome using the character features and linear-interpolation.

Nobata et al. (Nobata 1999) comment on the particular difficulties with identifying and classifying terms in the biochemistry domain including an open vocabulary and irregular naming conventions as well as extensive cross-over in vocabulary between classes. The irregular naming arises in part because of the number of researchers from different fields who are working on the same knowledge discovery area as well as the large number of proteins, DNA etc. that need to be named. Despite the best efforts of major journals to standardize the terminology, there is also a significant problem with synonymy so that often an entity has more than one name that is widely used such as the protein names AKT and PKB. Class cross-over of terms is another problem that arises because many DNA and RNA are named after the protein with which they transcribe.

Despite the apparent simplicity of the knowledge in NEHMM, the model has proven to be quite powerful in application. In the genome domain with only 80 training MEDLINE abstracts it could achieve over 74% F-score (a common metric for evaluation used in IE that combines recall and precision). Similar performance has been found when training using the dry-run and test set for MUC-6 (60 articles) in the news domain.

The next stage in the development of our model is to train using larger test sets and to incorporate wider contextual knowledge, perhaps by marking-up for dependencies of named-entities in the training corpus. This extra level of structural knowledge should help to constrain class assignment and also to aid in higher levels of IE such as event extraction.

7 Knowledge Building and Text Annotation

Annotated corpora constitute not only an integral part of a linguistic investigation but also an essential part of the design methodology for an NLP systems. In particular, the design of IE systems requires clear understanding of information formats of the domain, i.e. what kinds of entities and events are considered as essential ingredients of information. However, such information formats are often implicit in the minds of domain specialists and the process of annotating texts helps to reveal them.

It is also the case that the mapping between information formats and surface linguistic realization is not trivial and that capturing the mapping requires empirical examination of actual corpora. While generic programs with learning ability may learn such a mapping, learning algorithms need training data, i.e. annotated corpora.

In order to design a NE recognition program, for example, we have to have a reasonable amount of annotated texts which show in what linguistic contexts named entities appear and what internal structures typical linguistic expressions of named entities of a given field have. Such human inspection of annotated texts suggests feasible tools for NE (e.g. HMM, ME, decision trees, dictionary look-up, etc.) and a set of feasible features, if one uses programs with learning ability. Human in-

spection of annotated corpora is still an inevitable step of feature selection, even if one uses programs with learning ability.

More importantly, to determine classes of named entities and events which should reflect the views of domain specialists requires empirical investigation, since these often exist implicitly only in the mind of specialists. This is particularly the case in the field of medical and biological sciences, since they have a much larger collection of terms (i.e. class names) than, for example, mathematical science, physics, etc.

In order to see the magnitude of the work and difficulties involved, we chose a well-circumscribed field and collected texts (MEDLINE abstracts) in the field to be annotated. The field is the reaction of transcription factors in human blood cells. The kinds of information that we try to extract are the information on protein-protein interactions.

The field was chosen because a research group of National Health Research Institute of the Ministry of Health in Japan is building a database called CSNDB (Cell Signal Network DB), which gathers this type of information. They read papers every week to extract relevant information and store them in the database. IE of this field can reduce the work that is done manually at present.

We selected abstracts from MEDLINE by the key words of "human", "transcription factors" and "blood cells", which yield 3300 abstracts. The abstracts are from 100 to 200 words in length. 500 abstracts were chosen randomly and annotated. Currently, semantic annotation of 300 abstracts has been finished and we expect 500 abstracts to be done by April (Ohta 2000).

The task of annotation can be regarded as identifying and classifying the terms that appear in texts according to a pre-defined classification scheme. The classification scheme, in turn, reflects the view of the fields that biochemists have. That is, semantic tags we use are the class names in an ontology of the field.

Ontologies of biological terminology have been created in projects such as the EU funded GALEN project to provide a model of biological concepts that can be used to integrate heterogeneous information sources while some ontologies such as MeSH are built for the purpose of information retrieval According to their purposes, ontologies differ from fine-grained to coarse ones and from associative to logical ones. Since there is no appropriate ontology that covers the domain that we are interested in, we decided to build one for this specific domain.

The design of our ontology is in progress, in which we distinguish classification based on roles that proteins play in events from that based on internal structures of proteins. The former classification is closely linked with classification of events. Since classification is based on feature lattices, we plan to use the LiLFeS system to define these classification schemes and their relationships among them.

8 Future Directions

While the researches of the 80s and 90s in NLP focussed on different aspects of language, they have been so far considered separate development and no serious attempt has been made to integrate them.

In the JSPS project, we have prepared necessary background for such integration. Technological background such as efficient parsing, a programming system based on types, etc. will contribute to resolving efficiency problems. The techniques such as NE recognition, staged architecture in conventional IE, etc. will give hints on how to incorporate several different techniques in the whole system. A reasonable size of semantically annotated texts, together with relevant ontology, have been prepared.

We are engaged now in integrating these components in the whole system, in order to show how theoretical work, together with collection of empirical data, can facilitate systematic development of NLP application systems.

References

Collier, N., Nobata, C., and Tsujii, J.: "Extracting the Names of Genes and Gene products with

a Hidden Markov Model", COLING'2000 (August), 2000

Mitsuishi, Y. et.al.: HPSG-style Underspecified Japanese Grammar with Wide Coverage, in Proc. of Coling-ACL 98, Montreal, 1998

Miyao, Y., Makino, T., et.al.: The LiLFeS Abstract Machine and its Evaluation with LinGo, A Special Issue on Efficient Processing of HPSG, Journal of Natural Language Engineering, Cambridge University Press, 2000 (to appear)

Nobata, C., Collier, N., and Tsujii, J.: "Automatic Term Identification and Classification in Biology Texts", in proceedings of the Natural Language Pacific Rim Symposium (NLPRS'99), Beijing, China, 1999.

Ohta, T., et.al.: A Semantically Tagged Corpus based on an Ontology for Molecular Biology, in Proc. of JSPS Symposium 2000, Tokyo, 2000

Tateisi, Y. et.al.: Translating the XTAG English Grammar to HPSG, in Proc. of TAG+4 workshop, University of Pennsylvania, 1998

Torisawa, K. et.al.: An HPSG Parser with CFG Filtering, A Special Issue on Efficient Processing of HPSG, Journal of Natural language Processing, Cambridge University Press, 2000 (to appear)

Tsujii, J.: MT Research : Productivity and Conventionality of Language, RANLP-95,Tzigov Chark,Bulgaria,14-16 September,1995

Yakushiji, A.: Domain-Independent System for Event Frame Extraction using an HPSG Parser, Bsc Dissertation, Department of Information Science, University of Tokyo, 2000