Morphological Analysis from the Raw Kashmiri Corpus Using Open Source Extract Tool

Manzoor Ahmad Chachoo*
S. M. K. Quadri**

Abstract

Purpose: *Morphological information is a key part when we consider the design of any machine translation engine, any information retrieval system or any natural language processing application. It is important to investigate how lexicon development can be automated maintaining the quality which makes it of use for the applications, since manual development can be highly time consuming task. The paper describe how we can simply provide the extraction rules along with raw texts which can guide the computerized extraction of morphological information with the help of the extract tool like Extract v2.0.*

Design/methodology/approach: *We used Extract v2.0 which is an open source tool for extracting linguistic information from raw text, and in particular inflectional information on words based on the word forms appearing in the text. The input to the Extract is a file containing, an un-annotated Kashmiri corpus and a file containing the Extract rules for the language. The tools output is the list of analyses; each analysis consists of a sequence of words annotated with a identifier that describes some linguistic information about the word.*

Findings: *The study includes the fundamental extraction rules which can guide the Extract tool v2.0 to extract the inflectional information and help in the development of a full lexicon that can be use for developing different applications in the natural language applications. The major contributions of the study are:*

- ***Orthography component:*** *A Unicode Infrastructure to accommodate Perso-Arabic script of Kashmiri.*
- ***Morphology component****: A type system that covers the language abstraction and an inflection engine that covers word-and-paradigm morphological rules for all word classes.*

Research Implications: *The study however does not include all the rules but can be taken as a prototype for extending the functionality of the lexicon. An attempt has been made to make use of automated morphological information using Extract tool.*

Originality/Value: *Kashmiri language is the most widely spoken language in the state of Jammu and Kashmir. The language has very scarce software tools and applications. The study provides a framework for the development of a full size lexicon for the Kashmiri language from the raw text. The study is an attempt to provide a lexicon support for the applications which make use of Kashmiri language. This study can be extended for developing spoken lexicon of Kashmiri language that can be used in spoken dialogue systems.*

* *Faculty. P. G. Department of Computer Sciences, University of Kashmir, Jammu and Kashmir. 190 006. India. email: manzoor@kashmiruniversity.net*

** *Head. P. G. Department of Computer Sciences, University of Kashmir, Jammu and Kashmir. 190 006. India. email: quadrismk@hotmail.com*

Keywords: *Natural Language Processing; Morphology; Lexicon; Kashmiri Morphology; Extract Tool; Logic*
Paper Type: *Design*

Introduction

Morphological information is a key part when we consider the design of any machine translation engine, any information retrieval system or any natural language processesing application. It is important to investigate how lexicon development can be automated maintaining the quality which makes it of use for the applications since manual development can be highly time consuming task. Attempts have been made to use unsupervised learning to automate the process **(Forsberg & Ranta, 2004; Creutz & Lagus, 2005)** but if under the supervision of humans who simply have to provide knowledge about the rules along with raw texts can guide the computerized extraction of morphological information with the help of the extract tool. Extract v2.0 is an open source tool for extracting linguistic information from raw text, and in particular inflectional information on words based on the word forms appearing in the text. The input to the Extract is a file containing, an un-annotated Kashmiri corpus and a file containing the Extract rules for the language. The tools output is the list of analyses, each analysis consists of a sequence of words annotated with a identifier that describes some linguistic information about the word

Morphological lexicon with a wide coverage especially with new words as used in newspaper, texts and online sources forms a key requirement of the information retrieval systems, machine translation and other natural language applications. It would be a time consuming task to extract morphological information manually, so it is natural to investigate how the lexicon development can be automated. Since large collections of raw language data in form of technical texts, newspapers and online material are available and either free or cheap, it is an intelligent idea to exploit the raw data to obtain the high-quality morphological lexicon **(Forsberg & Ranta, 2004)**. Clearly, attempts to fully automatize the process using the

supervised learning technique do not return the quality as expected **(Creutz & Lagus, 2005; Sharma, Kalita & Das, 2002)**. However, instead of using different techniques of machine learning for lexicon extraction in some form, the language experts can use a suitable open source tool like Extract v2.0 wherein their role would be to write intelligent extraction rules. The extract tool will start with a large-sized corpus and a description of the word forms in the paradigms with the varying parts, referred to as technical stems, represented with variables. In the tool's syntax, we could describe the first declension noun of Kashmiri with the following definition.

paradigm decl1 =

x+"r"

{ x+"i" & x+"iv" & x+"I" & x+"in" } ;

All the forms are given in the curly braces , called the constraint, for some prefix x, the tool outputs the head x+"r" tagged with the name of the paradigm for example Ka:r can have other forms like Kar:iv , Ka:ri ,Kar:in.

Given that we have the lemma and the paradigm class label, it is a relatively simple task to generate all word forms. The paradigm definition has a major drawback: very few lemmas appear in all word forms but the tool a solution by supporting propositional logic in the constraint.

Related Work

The most important work dealing with the very same problem, i.e. extracting a morphological lexicon given a morphological description, is the study of the acquisition of French verbs and adjectives by **Cl´ement, Sagot & Lang (2004)**. Likewise, they start from an existing inflection engine and exploit the fact that a new lemma can be inferred with high probability if it occurs in raw text in predictable morphological form(s). Their algorithm ranks hypothetical lemmas based on the frequency of occurrence of its (hypothetical) forms as well as part of- speech information signaled from surrounding closed-class words. They do not make use of human-written rules but reserve an unclear, yet crucial, role for the human to hand-validate parts of output and then let the algorithm

re-iterate. Given the many differences, the results cannot be compared directly to ours but rather illustrate a complementary technique.

Tested on Russian and Croatian, **Oliver (2004); Oliver and Tadic (2004 a)** describe a lexicon extraction strategy very similar to ours. In contrast to human-made rules, they have rules extracted from an existing (part of) a morphological lexicon and use the number of inflected forms found to heuristically choose between multiple lemma-generating rules (additionally also querying the Internet for existence of forms). The resulting rules appear not at all as sharp as hand-made rules with built-in human knowledge of the paradigms involved and their respective frequency (the latter being crucial for recall). Also, in comparison, our search engine is much more powerful and allows for greater flexibility and user convenience. For the low-density language Assamese, **Sharma, Kalita & Das (2002)** report an experiment to induce both morphology, i.e. the set of paradigms, and a morphological lexicon at the same time. Their method is based on segmentation and alignment using string counts only – involving no human annotation or intervention inside the algorithm. It is difficult to assess the strength of their acquired lexicon as it is intertwined with induction of the morphology itself. We feel that inducing morphology and extracting a morphological lexicon should be performed and evaluated separately. Many other attempts to induce morphology, usually with some human tweaking, from raw corpus data **(Goldsmith, 2001)**, do not aim at lexicon extraction in their current form.

There is a body of work on inducing verb sub categorization information from raw or tagged text **(Faure & Nedellec, 1998; Gamallo, Agustini & Lopes, 2003; Kermanidis, Nikos & Kokkinakis, 2004)**. However, the parallel between sub categorization frame and morphological class is only lax. The latter is a simple mapping from word forms to a paradigm membership, whereas in verb sub categorization one also has the onus discerning which parts of a sentence are relevant to a certain verb. Moreover, it is far from clear that verb sub categorization comes in well-defined paradigms – instead the goal may be to reduce the amount of

parse trees in a parser that uses the extracted sub categorization constraints.

Methodology

Kashmiri is a mix of both agglutinating and inflectional type of language. Agglutinating language consists of poly morphemic words in which each morpheme corresponds to a single lexical meaning or grammatical function and by inflectional means that the lexical meanings and grammatical functions are at times fused together. Morphemic processes across most lexical categories such as nouns, verbs, adjectives and adverbs are studied and converted into rules which are input to the extract tools e.g.

Nouns in Kashmiri are not marked for being definite. There is an optional indefinite marker –a:h

Also animate nouns follow the natural gender system. Gender of a large number of inanimate nouns is predictable from their endings.

The following suffixes are added to nouns to derive masculine forms : -da:r, -dar , -vo:l, -ul and –ur

paradigm decl2 =

x+"r"

{ x+" da:r " & x+"-dar " & x+"-vo:l " & x+"-ul " } ;

The following suffixes are added to nouns to derive feminine forms : -en, -in , -e:n, --ba:y , -ir and –va:jen

paradigm decl3 =

x+"r"

{ x+" en " & x+"- in " & x+"-e:n " & x+"-ir " & x+"-ir " } ;

Morphology

Morphology is the study of morphemes, and Morphemes are words, word stems, and affixes, basically the unit of language one up from phonemes. These are often understood as units of meaning, and also part of a language's syntax or grammar.

It is in their morphology that we most clearly see the differences between languages that are ***isolating*** (such as *Chinese, Indonesian,*

Krewol...), ones that are ***agglutinating*** (such as *Turkish, Finnish, Tamil*...), and ones that are ***inflexional*** (such as *Kashmiri, Russian, Latin, Arabic*...). Isolating languages use grammatical morphemes that are separate words. Agglutinating languages use grammatical morphemes in the form of attached syllables called affixes. Inflexional languages change the word at the phonemic level to express grammatical morphemes.

All languages are really mixed systems -- it's all a matter of proportions. English, for example, uses all three methods: To make the future tense of a verb, we use the particle will *(I will see you)*; to make the past tense, we usually use the affix -ed *(I changed it)*; but in many words, we change the word for the past *(I see it becomes I saw it)*. Looking at nouns, sometimes we make the plural with a particle *(three head of cattle)*, sometimes with an affix *(three cats)*, and sometimes by changing the word *(three men)*. But, because we still use a lot of non-syllable affixes (such as -ed, usually pronounced as d or t, and -s, usually pronounced as s or z, depending on context), English is still considered an inflexional language by most linguists.

Paradigm File Format

A paradigm file consists of two kinds of definitions: regexp and paradigm. A regexp definition associates a name (Name) with a regular expression (Reg). A paradigm definition consists of a name (Name), a set of variable regular expression associations (VarDef), a set of output constituents (Head) and a constraint (Logic). The basic unit in Head and Logic is a pattern that describes a word form. A pattern consists of a sequence of variables and string literals glued together with the '+' operator. An example of a pattern given previously was x+"r".

Propositional Logic

Propositional logic appears in the constraint to enable a more fine-grained description of what word forms the tool should look for. The basic unit is a pattern, corresponding to a word form, which is combined with the operators & (and), | (or), and ~ (not). The syntax for

propositional logic is given in **Fig. 1**, where Pattern refers to one word form.

Fig. 1: Propositional logic grammar

```
kLog ::= kLog & kLog
| kLog | kLog
| kLog
| ~ kLog
| kPattern
| ( kLog )
```

The addition of new operators allow the paradigm in section 1 to be rewritten with disjunction to reflect that it is sufficient to find one singular and one plural word form. The middle vowel /o/ of the structure nouns changes to a central vowel and the final consonant is palatalized.

```
paradigm decl1 =
x+"r"
{ (x+"l" | x+"ur") } ;
```

Regular Expressions

The variable part of a paradigm description provided by the tool is to enable the user to associate every variable with a regular expression. The association dictates which (sub-) strings a variable can match. An unannotated variable can match any string, i.e. its regular expression is Kleene star over any symbol. As a simple example, consider German, where nouns always start with an
uppercase letter. This can be expressed as follows.

```
regexp UpperWord = upper letter*;
paradigm n [x:UpperWord] = ... ;
```

The syntax of the tool's regular expressions is given in **Fig. 2**, with the normal connectives: union, concatenation, set minus, Kleene star, Kleene plus and optionality. eps refers to the empty string, digit to 0 − 9, letter to an alphabetic Unicode character, lower and upper to a lowercase respectively an uppercase letter. char refers to any character. A regular

expression can also contain a double-quoted string, which is interpreted as the concatenation of the characters in the string.

Fig. 2: Regular expression

```
kReg ::= kReg | kReg
       | kReg – kReg
       | kReg kReg
       | kReg *
       | kReg +
       | kReg ?
       | eps
       | kChar
       | digit
       | letter
       | upper
       | lower
       | char
       | kString
       | ( kReg )
```

Multiple Variables

The Extract tool allows multiple variables, i.e. a pattern may contain more than one variable.

The use of variables may reduce the time-performance of the tool, since every possible variable binding is considered. The use of multiple variables should be moderate, and the variables should be restricted as much as possible by their regular expression association to reduce the search space.

A variable does not need to occur in every pattern, but the tool only performs an initial match with patterns containing all variables. The reason for this is efficiency — the tool only considers one word at the time, and if the word matches one of the patterns, it searches for all other patterns with the variables instantiated by the initial match. For obvious reasons, an initial match is never performed under a negation, since this would imply that the tool searches for something it does not want to find.

It is allowed to have repeated variables, i.e. non-linear patterns, which is equivalent to back reference in the programming language Perl. An example where a sequence of bits is reduplicated is given. This language is known to be non-context-free **(Hopcroft & Ullman, 2001)**.

```
regexp ABs = (0|1)*;
paradigm reduplication [x:ABs] =
x+x { x+x } ;
```

Multiple Arguments

The head of a paradigm definition may have multiple arguments to support more abstract paradigms. An example is of Swedish nouns, where many nouns can be correctly classified by just detecting the word forms in nominative singular and nominative plural. An example is given **(Fig. 3)**, where the first and second declension is handled with the same paradigm function, where the head consists of two output forms. The constraints are omitted.

Fig. 3

```
paradigm regNoun = paradigm regNoun =
gag+"ar" gag+"ir" kot+"ur" ko+":tar"
{...} ; {...} ;
```

The Algorithm

Fig. 4 represents the algorithm of the tool is presented in pseudo-code notation.

Fig. 4

```
let L be the empty lexicon.
let P be the set of extraction paradigms.
let W be all word types in the corpus.
for each w : W
for each p : P
for each constraint C with which w matches p
if W satisfies C with the result H,
add H to W
endif
end
end
end
```

The algorithm is initialized by reading the word types of the corpus into an array W. A word w matches a paradigm p, if it can match any of the patterns in the paradigm's constraint that contains all variables occurring in the constraint. The result of a successful match is an instantiated constraint C, i.e. a logical formula with words as atomic propositions. The corpus W satisfies a constraint C if the formula is true, where the truth of an atomic proposition ***"a"*** means that the word ***"a"*** occurs in W.

Conclusion

The paper describes the open source extract tool as a means to build morphological lexicon which requires relatively less human work. Given a morphological description, typically an inflection engine and a description of the closed word classes, such as pronouns and prepositions, and access to raw text data, a human with knowledge of the language can use a simple but versatile tool that exploits word forms alone. It remains to be seen to what extent syntactic information, e.g. part-of-speech information, can further enhance the performance. A more open question is whether the suggested approach can be generalized to collect linguistic information of other kinds than morphology, such as e.g. verb sub categorization frames.

References

Forsberg, M., & Ranta, A. (2004). Functional Morphology. In *Proceedings of the ninth ACM SIGPLAN international conference on Functional programming (ICFP '04) (pp. 213-223)*. *Snow Bird UT, U.S.A.* New York: ACM. doi: 10.1145/1016850.1016879

Creutz, M., & Lagus, K. (2005). Inducing the morphological lexicon of a natural language from unannotated text. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR '05), 15-17 June, Espoo, Finland, Espoo (2005) (106–113). Espoo, Finland*. Retrieved from

http://research.ics.tkk.fi/events/AKRR05/papers/akrr05creutz.pdf

Sharma, U., Kalita, J., & Das, R. (2002). Unsupervised learning of morphology for building lexicon for a highly inflectional language. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning (MPL '02) (pp. 1-10). Stroudsburg, PA, USA: Association for Computational Linguistics*. doi: 10.3115/1118647.1118648

Hopcroft, J. E., & Ullman, J. D. (2001). *Introduction to automata theory, languages, and computation (2nd ed.)*. Reading, Mass: Addison-Wesley.

Cl´ement, L., Sagot, B., & Lang, B. (2004). Morphology based automatic acquisition of large-coverage lexica. Retrieved from http://hal.archives-ouvertes.fr/docs/00/41/31/89/PDF/LREC04.pdf

Oliver, A. (2004).*Adquisici´o d'informaci´o l`exica i morfosint`actica a partir de corpus sense anotar: aplicaci´o al rus i al croat*. (PhD Thesis). Universitat de Barcelona.

Oliver, A., & Tadic, M. (2004 a). Enlarging the croatian morphological lexicon by automatic lexical acquisition from raw corpora. In *Proceedings of LREC'04, Lisboa, Portugal (2004) 1259–1262*. Retrieved from http://www.hnk.ffzg.hr/txts/aomt4lrec2004.pdf

Goldsmith, J. (2001). Unsupervised learning of the morphology of natural language. *Computational Linguistics*. 27(2), 153–198. doi: 10.1162/089120101750300490

Kermanidis, K. L., Nikos, F., & Kokkinakis, G. (2004). Automatic acquisition of verb subcategorization information by exploiting mininal linguistic resources. *International Journal of Corpus Linguistics*, 9 (1), 1-28. doi: 10.1075/ijcl.9.1.01ker

Faure, D., & Nedellec, C. (1998). Asium: Learning subcategorization frames and restrictions of selection. In Y. Kodratoff (Ed.). *10th Conference on Machine Learning (ECML 98) – Workshop on Text*

Mining, Chemnitz, Germany, Avril 1998. Springer-Verlag, Berlin (1998)

Gamallo, P., Agustini, A., & Lopes, G.P. (2003). Learning subcategorisation information to model a grammar with "Co-restrictions". *Traitement Automatique des Langues.* 44 (1), 93–177