Building NLP resources for Dzongkha: A Tagset and A Tagged Corpus

Chungku Chungku, Jurmey Rabgay
Research Division
Department of Information Technology & Telecom
{chungku,jrabgay}@dit.gov.bt

Gertrud Faaß
Institute für Maschinelle Sprachverarbeitung(NLP processing),
University of Stuttgart
faasz@ims.uni-stuttgart.de

Abstract

This paper describes the application of probabilistic part of speech taggers to the Dzongkha language. A tag set containing 66 tags is designed, which is based on the Penn Treebank[1]. A training corpus of 40,247 tokens is utilized to train the model. Using the lexicon extracted from the training corpus and lexicon from the available word list, we used two statistical taggers for comparison reasons. The best result achieved was 93.1% accuracy in a 10-fold cross validation on the training set. The winning tagger was thereafter applied to annotate a 570,247 token corpus.

1 Introduction

Dzongkha is the national language of Bhutan. Bhutan has only begun recently applying Natural Language Processing (henceforth NLP) methodologies and tools. However, Dzongkha computing is currently progressing rapidly.

Part of speech (henceforth POS) tagging means annotating each word with their respective POS label according to its definition and context. Such annotation generates a description of the text on a meta-level, i.e. a representation of linguistic units on the basis of their properties. This POS-level provides significant information usable by further linguistic research, may it be of the morphological, syntactic or semantic kind. Producing such enriched data is proven to be useful especially when designing NLP representations of higher levels of representation, e.g. syntactic parses.

Our project is designed to annotate Dzongkha cyclopedia text with parts of speech using a probabilistic tagger. This means that a set of tags is to be developed and applied manually to parts of these texts creating training data. Probabilistic taggers can then be applied to annotate other texts with their parts of speech automatically. In this paper, we make use of two such taggers and report on their results.

At present, our POS tagged data is already in use in projects concerning Dzongkha Text to Speech (TTS) processing, further tests on word segmentation (see current state below) and in corpus-linguistic research. Future work entails its utilization for higher-level NLP tasks such as parsing, building parallel corpora, research on semantics, machine translation, and many more.

Sections 2 and 3 of this paper describe the Dzongkha script and the challenges in Dzongkha, section 4 presents our resources, tagset and corpus. Section 5 describes tagging and validation processes and reports on their results. Section 6 concludes and discusses future work.

2 The Dzongkha Language

Dzongkha is recognized as the national and official language of Bhutan. It is categorized as a Sino-Tibetan Language and said to have derived from the classical Tibetan or choka: Dzongkha consonants, vowels, phonemes, phonetics and writing system are all identical.

[1] [http://www.cis.upenn.edu/~treebank/]

Proceedings of the 8th Workshop on Asian Language Resources, pages 103–110,
Beijing, China, 21-22 August 2010.

From a linguistic perspective, Dzongkha script is syllabic, a syllable can contain one character or as many as six characters. A syllable marker known as “tsheg”, which is simply a superscripted dot, separates the syllables of a word. Linguistic words may contain one or more syllables and are also separated by the same symbol, “tsheg”, thus the language is lacking word boundaries.

Sentences of Dzongkha contain one or more phrases which themselves contain one or more words. A character known as “shed” marks a sentence border, it looks like a vertical pipe.

Phonetic information is available, too: In most sentences, a pause of silence is taken after each phrase while speaking the Dzongkha language. The written form of Dzongkha represents this pause with a space after each phrase in the case that it occurs not at the end of the sentence. The Dzongkha writing system leads to a serious problem: the detection of word borders, because only phrases are separated by a space. POS tagging usually requires a one-token-per line format, which is produced by a process called word segmentation. The tagger then adds the POS category to each token.

The training data of (40247 tokens) was segmented manually to achieve higher accuracy of word boundary and also due to lack of word segmentation during that time. After a Dzongkha word segmentation[2] tool was developed, the remaining text was segmented with this tool, which works basically with a lexicon and the longest string matching method.

3 Challenges and suggestions for tagging Dzongkha texts

3.1 Words unknown to the language model

A statistical tagger learns from POS distributions in manually tagged data while being trained and, when being applied to unknown text, “guesses” the POS of each word. The TreeTagger (Schmid, 1994) additionally makes use of a lexicon externally provided when producing its language model (the “parameter file”). We had opted for using the TreeTagger and hence we have listed about 28,300 Dzongkha words with their POS in a lexicon selected from the 570,247 token corpus to be tagged. We fed these data to the tagger during its training phase. Note, however, that such a lexicon may never be complete, as there are morphologically productive processes creating new forms (these belong to POS classes that are often named “open”). Such forms may be taken into account when developing a tagset, however, in this work, we opted for postponing the issue until a morphological analyser can be developed.

3.2 Ambiguous function words

A number of Dzongkha function words are ambiguous; for each occurrence of such a word, the tagger has to decide on the basis of the word’s contexts, which of the possible tags is to be assigned. Here, the tagset itself comes into view: whenever it is planned to utilize probabilistic POS taggers, the tagset should be designed on the basis of the words’ distributions, otherwise the potential accuracy of the taggers may never be achieved.

In Dzongkha it is mainly the function words that are ambiguous in terms of their POS. A typical example is ལས་/le/(from) belonging to the category PP (post position) and ལས་/le/(so) which is of the category CC (conjunction).

3.3 Fused forms

Some morpho-phonemic processes in Dzongkha lead to the fusing of words, presenting another challenge for tagging. Such words[3] are not very frequent, thus proposing a challenge to statistical taggers. The word རྒྱལ་པོའི་/gelpoi/ (king$_{[+genitive]}$), for example, is fused from the phrase རྒྱལ་པོ་གི་/gelpo gi/ (king is); another example is the fused form གསེད་/sen/ ([to] kill), made from གསེད་བཅིན་/se wa cin/ (if [to] kill).

When a tagset does not cater for fused forms of words, one could split these forms while tokenizing adding an intermediate level of representation between original text level and

[2]This tool was developed at NETEC(National Electronics and Computer Technology Center), Thailand.

[3] In our training set, there were 1.73% of all words detected as fused forms.

the POS level: a level of text material to be utilized for tagging or other further processing, as e.g. done by (Taljard et al., 2008) for the Bantu Language Northern Sotho. However, the forms could not easily be split, as the resulting first parts of the words would not contain the separator “tsheg”. Splitting the word རྒྱལ་པོའི་/gelpoi/ (king$_{[+genitive]}$), for example, would result in རྒྱལ་པོ/gelpo/(king) and འི་/yi/$_{[+genitive]}$. The language model does not cater for words ending in characters other than "tsheg" (word border) or being directly followed by "shed" (a word like རྒྱལ་པོ/gelpo/(king) may only appear if preceding a sentence border). Tagging accuracy for such theoretical forms are not expected to be acceptable. Fusing processes are productive, therefore, further research in the frame of a project developing a Dzongkha tokenizer is deemed necessary.

We examined all fused words contained in our textual data to find an workable solution at the current stage of the project. As long as the problem of tokenizing automatically is not solved, we opted for keeping the fused forms as they are. To enhance our tagger results, we suggest to add a number of tags to our tagset that consist of the two POS tags involved. རྒྱལ་པོའི་/gelpoi/ (king$_{[+genitive]}$), for example, is tagged as “NN+CG” and བསད་/sen/ ([to] kill) as “VB+SC”. The “+” indicates combined individual tags. All known forms are added to a new version of the lexicon. Note, however, that all tagging results reported upon in this paper, are still based on the tag set described below.

4 Resources used

4.1 Tagset

During the first phase of PAN Localization project, the first Dzongkha POS tagset[4] was created. It consisted of 47 tags, its design is based on the Penn Guidelines[5] and its categories of POS correspond to the respective English Penn categories. PAN generally makes use of the Penn Treebank tags as a basis for tagging. Examining the similar features exhibited by both the languages (Dzongkha and English), tags that were applicable to Dzongkha were taken directly from the Penn Treebank. In cases where these languages showed dissimilarities in their nature, new tags for Dzongkha were assigned (based e.g. on the work on Urdu of Sajjad and Schmid, 2009). As an example for such dissimilarity, Dzongkha postpositions are mentioned here, cf. (1); the respective tag (PP) only exist for Dzongkha whereas in English the whole set of ad position tags (preposition and postpositions) exist.

(1) ཇི་ཡི་ ཤིང་གི་ཝོ་ལུ་ལྡོག

i'ili	shing-gi	wôlu	-dû
Cat	tree$_{[noun]}$	under$_{[PP]}$	be

"A cat is under the tree"

Whenever a tagset designed on theoretical implications is applied to text, it will be found in the course of creating the training data that not all morpho-syntactic phenomena of the language had been considered. This happened for Dzongkha, too: words appeared in the texts that didn't fit in any of the pre-defined classes.

Dzongkha uses honorific forms: ན་བཟའ་/nam za/ (cloths) is the honorific form of the noun གོ་ལ་/gola/(cloths), གསུངས་/sung/(tell) the honorific form of the verb ལྟབ་/lab/(tell). We opted to mark them by adding the tag NNH (honorific common noun) and VBH (honorific verb) to enable future research on this specific usage of Dzongkha language. A number of tags were added to the set, of which we describe four in more detail: two of the additional tags are sub-classes of verbs: VBH (honorific verb form), and VBN which describes past participle forms, like, e.g. ཇུངས་/jun/(created), the past particle form of ཇུང་/jung/(create).

Concerning case, we added two subclasses of case: CDt and CA. These differentiate between dative (CDt) and ablative (CA): The CDt (Dative case) labels e.g.དོན་ལས་/doen le/(for it) and དོན་ལུ་/doen lu/(for this). The Ablative case

[4] The original Dzongkha tag set is described at http://www.panl10n.net

[5] The Penn Guidelines can be downloaded from: http://www.cis.upenn.edu/~treebank/

(CA) is used when the argument of the preposition describes a source. For example, in the phrase ཤིང་ལས་ཀྲང་ཐྲི་/shing le kang thri/(from wood chair), ལས་/le/from/ will be labeled CA since the chair described is made from (the source) wood (Muaz, et al. 2009). The tagset utilized in our experiment consists of a total of 66 parts of speech as shown in Appendix (A).

4.2 Collecting a Corpus and generating a training data set

The Corpus collection process. The process of collecting a corpus should be based on its purpose. As our goal was the design a Dzongkha text corpus as balanced as possible in terms of its linguistic diversity, the text data was gathered from different sources like newspaper articles, samples from traditional books, and dictionaries, some text was added manually (poetry and songs). The text selection was also processed with a view on the widest range of genres possible: texts from social science, arts and culture, and texts describing world affairs, travel adventure, fiction and history books were added as our goal is to make it representative of every linguistic phenomena of language (Sarkar, et al. 2007). The corpus is however not balanced for a lack of available electronic resources of informative text (so far only 14% belong to this category). Future work will therefore entail collecting more data from respective websites and newspapers.
The entire corpus contains 570,247 tokens; it made from the domains described in table (1).

Domain	Share %	Text type
1) World Affairs	12%	Informative
2) Social Science	2%	Informative
3) Arts	9%	Descriptive
4) Literature	72%	Expository
5) Adventure	1%	Narrative
6) Culture	2%	Narrative
7) History	2%	Descriptive

Table (1): Textual domains contained in the corpus

The Training data set

Cleaning of texts. Raw text is usually to be modified before it can be offered to a tagger. It is to be cleaned manually, e.g. by removing extra blank spaces, inserting missing blanks, correcting spelling mistakes, and by removing duplicate occurrences of sequences. Secondly, the process of tokenization ("Word Segmentation") is to be applied.

Design and generation of training data. The training data set was produced in several steps: Firstly, 20,000 tokens were manually labeled with their respective parts of speech (for a comparison of tagging techniques, cf. Hasan et al., 2007). Thereafter, the problems that had occurred during the manual process were summarized and the tagset revised as described in section 4.1. Thereafter, we added another 20,247 tokens. The final training data set hence consists of 40,247 tokens (2,742 sentences, 36,362 words, 3,265 punctuation, 650 numbers).

4.3 Tagging technique: TreeTagger and TnT

TreeTagger (Schmid, 1994): TreeTagger (Schmid, 1994) is a probabilistic part of speech tagger operating on the basis of decision trees. Helmut Schmid developed it in the frame of the "TC"[6] project at the Institute for Computational Linguistics at the University of Stuttgart, Germany.

The software consists of two modules

a) train-tree-tagger: utilized to generate a parameter file from a lexicon and a hand-tagged corpus.

b) tree-tagger: makes use of the parameter file generated with a); annotates text (which is to be tokenized first) with part-of-speech automatically.

a) Generating a language model: Training

When generating a language model stored in a so-called "parameter file", three files are required: a lexicon describing tokens and their respective tags, a list of open tags, and training

[6] The tagger is freely available at http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTreeTagger.html

data. The “train-tree-tagger” module generates a binary parameter file.

The lexicon is a file that contains tokens (word forms and punctuation), in the format of one per line. The TreeTagger was developed for languages with inflection, i.e. languages where one word may occur in a number of allomorphs. To ease the task of tagging such languages, the tool can operate on the level of a base form, the “lemma” which may be added to every word form. In the case of Dzongkha, lemmatization has not been developed yet, therefore, we either make use of the word form itself, or a hyphen in the case that no lemma is available. As table (2) demonstrates, the lexicon contains the word forms in the first column, the second column contains the POS and the third a hyphen. In the case of ambiguous entries, one line may contain a sequence of tag-“lemma” pairs that follow the word form of the first column. The lexicon may not contain any spaces; all columns must be separated by exactly one tab(ulator). Because the lexicon is only made use of during the training phase of the tagger, any update must result in reproducing the parameter file.

Word	Pos tag	lemma	Pos tag	lemma
ཀ་	NN	ཀ་		
ཀ་ཀ་	NN	ཀ་ཀ		
ཀ་གུ་རུ་	NNP	--		
འམ་	PP	འམ་	CC	འམ་

Table (2) Example entries of the lexicon

Open class tags: a file containing the list of open class tags, i.e. the productive classes (one entry per line), cf. Appendix A. In the upcoming version of the tagset, tags of the following fused forms will be added, like, e.g. NN+CG (combination of all forms nouns with genitive case CG), VB+CG (combination of all forms verb with genitive case CG), JJ+CG (combination of all forms of adjective with genitive case CG), RB+CG (combination of all adverb with genitive case CG), and same with the combination of Subordinate conjunction NN+SC, VB+SC, JJ+SC, RB+SC, just to name a few.

Tagged training data: a file that contains tagged training data. The data must be stored in one-token-per-line format. This means that each line contains one token and its respective tag, these are separated by one tabulator. The file should be cleaned from empty lines, no meta-information, like, e.g. SGML markup is allowed.

b) Tagging

Two files serve as input: the binary parameter file and a text file that is to be tagged.

Parameter file: the file that was generated by step a) above.

Text file: a file that is to be tagged; it is mandatory that the data in this file appears in a one-token-per line format.

TnT (Brants, 2000):The Trigram’s’n’Tags (TnT) tagger was developed by Thorsten Brants (Brants, 2000). It is language independent and has been used widely for a number of languages, often yielding an accuracy of +96% without utilizing an external lexicon or an open word class file. TnT is based on Markov models, and takes not only distributional data of tokens, but also the final sequences of characters of a token into account when guessing the POS of a word. It can use the same format for training data as the TreeTagger, therefore, in order to use TnT for comparison reasons, no additional preparations for tagging Dzongkha are necessary.

5 Validation and Results

5.1 k-fold cross validation and bootstrapping

When applying a tagset to training data for the first time, it is advisable to progress in steps and to validate each step separately: One begins with annotating a rather small portion of text that is then divided into k number of slices. Slices k-1 are then utilized to create a parameter file, the slice k is stripped of its annotations and annotated by the tagger using that parameter file. The same procedure is followed for all other slices (“k-fold cross validation”).

Afterwards, a comparison between the original tags with the tags assigned by the tagger will then help to judge upon a number of issues, like, e.g., whether the size of the training data is sufficient (quantitative review). Examining the most frequent (typical) assignment errors of the tagger will also support the enhancement of the tagset: if e.g. the distribution of two different tags is more or less identical, a probabilistic tagger will not succeed in making the right choices, here, one is to consider if using one tag would be acceptable from a linguistic point of view (qualitative review).
The knowledge gained here usually leads to updates in the tagset and/or to the necessity to add more amounts of texts containing constellations that were found as being problematic for probabilistic tagging for they occur too rarely in the texts. After such updates are done on the existing training texts and tagset respectively, the k-fold validation may be repeated and reviewed again.
Updating training data and tagset will be repeated until the tagging results are satisfying (such a progressing method is usually called "bootstrap-ping").

5.2 TreeTagger results

The work on automatic part of speech tagging for Dzongkha began with the manual annotation of 20,000 tokens. Because a non-linguistic person performed the process manually, the language coordinator did thorough correction.
The 20,000 token training set, made use of 43 different single tags (of 47 provided by the tagset). The token-tag combinations from there were combined with an external lexicon produced from a dictionary; the resulting lexicon file thus contained all types.
The 10-fold cross validation resulted in an accuracy of around 78%. Result introspection lead to the knowledge that more data had to be added and that fused words will have to receive separate tags. It also showed that manual tokenization is an error-prone procedure, as a significant number of word and sentence borders had to be corrected in the data.
After updating tagset and training data, another 20,247 tokens were added to the training set and the lexicon was updated accordingly, except for the fused forms, where a final solution on how to tag them is not found yet. The tagset was extended to 66 tags (cf. Appendix A). With a full knowledge of the possible tag-token combinations, the Tree-Tagger achieved a median accuracy of 93.1%.

5.3 TnT results and comparison with the TreeTagger

Using the 40,247 tokens text segment, a 10-fold cross validation was also performed with the TnT tagger. It achieves a 91.5 % median accuracy when the tagset containing 47 tags is applied. Results for each slice and mean/median can be found in table (3) of both taggers for comparison reasons. TnT reports on the number of unknown tokens detected in each slice; the mean of 16.49 % (median 14.18%) of unknown tokens offers an explanation why TnT does not perform as good as the TreeTagger which was supplied with a complete lexicon thus not being faced with unknown tokens at all.

Tagger:	Tree-Tagger accuracy %	TNT accuracy %
slice 1	92.13	92.33
slice 2	84.61	89.73
slice 3	89.08	89.88
slice 4	90.17	90.43
slice 5	92.95	91.01
slice 6	93.32	91.35
slice 7	94.24	91.69
slice 8	93.32	92.03
slice 9	95.21	92.55
slice 10	94.56	92.60
Mean	91.96	91.36
Median	93.14	91.20

Table (3) 10-fold cross validation results for TreeTagger and TnT

A qualitative review of the results showed that usually it is the tag CC that is confused with others (NN, DT, NN, DT, PRL, etc.) by TnT, while the TreeTagger is rather confusing NN (with VB, NNP, PRL, CC).
However, a more thorough qualitative examination of these results is still to be done and may lead to further updates on the tagset.

6 Discussion, Conclusions and Future work

This paper describes the building of NLP resources for the national language of Bhutan, namely Dzongkha. We have designed and built an electronic corpus containing 570,247 tokens belonging to different text types and domains.
Automated word segmentation with a high precision/recall still remains a challenge. We have begun to examine statistical methods to find solutions for this and we plan to report on our progress in the near future.
We have developed a first version of a tag set on the basis of the Penn Tree tagset for English (cf. section 4.1) A training data set of 40,247 tokens has been tagged manually and thoroughly checked. Lastly, we have tagged the corpus with the TreeTagger (Schmid, 1994) using a full form lexicon achieving 93.1% and, for comparison reasons, with TnT (Brants, 2000), without a lexicon, achieving 91.5 %.
We have used the present output in the construction of an advance Dzongkha TTS (text to speech) using an HMM-based method which is developed by the Bhutan team in collaboration with HLT team at NECTEC, Thailand[7]
Loads of work still remains, we are still to examine the tagger results from a qualitative aspect in order to answer inter Alia the following questions: Are there any further updates on the tag set necessary, what is the best way to process fused forms. Quantitative aspects might also still play a role: It still might be necessary to add further training data containing part of speech constellations that rarely occur, so tagger results for those will enhance.
We also plan to increase our corpus collection from various ranges of domains. At present there are more media, e.g. newspapers available in the world wide web, we will be able to collect such texts easily. In Bhutan, there is an ongoing project on OCR (optical character recognition) of Dzongkha under the PAN project (www.PANL10n.net). Given the success of this project, we will be able to scan text from textbooks.

[7] http://www.nectec.or.th/en/

Acknowledgments

This research work carried out as part of PAN Localization Project (www.PAN10n.net) with the aid of grant from the International Development Research Center (IDRC), Ottawa, Canada, administered through the Center of Research in Urdu language Processing (CRULP), National University of Computer and Emerging Sciences (NUCES), Pakistan. The research team would also like to thank PD Dr. Helmut Schmid and Prof. Dr. Heid, Insitut für Maschinelle Sprachverarbeitung (NLP institute), Universität Stuttgart, Germany for their valuable support and contributions that made this research successful.

References

Brants, Thorsten. 2000. TnT - as statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-2000)*, Seattle, WA, USA, pages 224 – 231.

Hasan, Fahim Muhammad, Naushad UzZaman, and Mumit Khan. 2007. Comparison of different POS Tagging Techniques (N-Gram, HMM and Brill's tagger) for Bangla. *PAN Localization Working Papers*, 2004-2007, pages 31-37.

Hassan, Sajjad and Helmut Schmid . 2009. Tagging Urdu Text with Parts of Speech: A Tagger Comparison, *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)* . Athens, Greece, 2009.

Muaz, Ahmed, Aasim Ali, and Sarmad Hussain. 2009. *Analysis and Development of Urdu POS Tagged Corpus*. Association for Computational Linguistics. Morristown, NJ, USA. Retrieved December 1, 2009, from http://www.lancs.ac.uk

Schmid, Helmut. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*. Manchester, UK, pages 44 – 49.

Sarkar, Asif Iqbal, Dewan Shahriar Hossain Pavel and Mumit Khan. 2007. *Automatic Bangla Corpus Creation.* BRAC University, Dhaka, Bangladesh.

PAN Localization Working Papers, 2004-2007. pages 22-26.

Taljard Elsabé, Faaß Gertrud, Heid Ulrich, and Daan J. Prinsloo. 2000. On the development of a tagset for Northern Sotho with special reference to standardization. *Literator 29(1)*, April 2008 (special edition on Human Language Technologies), South Africa, pages 111 – 137

APPENDIX A

The Dzongkha Tagset
as used for the validation tests

Type	SubClass	Label
Open classes:		
Noun	Common Noun	NN
	Honorific form	NNH
	Particular/Person	NNP
	Quantifier	NNQ
	Plural	NNS
Verb	Aspirational	VBAs
	Honorific	VBH
	Agentive	VBAt
	Non-Agentive	VBNa
	Auxiliary	VBAUX
	Imperative	VBI
	Modal	VBMD
	Past participle	VBN
	Verb	VB
Adjective	Characteristic	JJCt
	Periodic	JJP
	Comparative	JJR
	Superlative	JJS
	Adjective	JJ
Adverb	Behavioral	RBB
	Comparative	RBR
	Superlative	RBS
	Adverb	RB
Interjection		UH
Closed classes:		
Marker	Affirmative	AM
	Interrogative	IrM
	Tense	TM
Case marker	Ablative Case	CA
	Dative Case	CDt
	Genitive Case	CG
	Vocative Case	CV
Pronouns	Locative	PRL
	Differential	PRD
	Personal	PRP
	Reflexive	PRRF
Conjunction	Coordinate	CC
	Subordinate	SC
Number	Cardinal Number	CD
	Ordinal Number	OD
	Nominal Number	ND
Ad position	Post position	PP
Determiner	Definite	DT
	Possessive	DT$
	Indefinite	DTI
Negator		NEG
Punctuation		PUN
Combined tags:		
Noun+Genitive case(CG)	Common+CG	NNCG
	Particular+CG	NNPCG
	Quantifier+CG	NNQCG
	Plural+CG	NNSCG
Adjective+CG	Adjective+CG	JJCG
	Characteristic +CG	JJCtCG
	Periodic+CG	JJPCG
Verb+CG	Honorific+CG	VBHCG
	Agentive+CG	VBAtCG
	Verb+CG	VBCG
	Modal+CG	VBMDCG
DefiniteDeterminer+CG	Determiner+CG	DTCG
Locative Pronoun +CG	Locative+CG	PRLCG
Negator+CG	Negator+CG	NEGCG
Noun+ Subordinate Conjunction(SC)	Common Noun +SC	NNSC
Verb+SC	Verb+SC	VBSC
	Agentive+SC	VBAtSC
	Modal verb+SC	VBMDC
Affirmative +SC	Affirmative +SC	AMSC
Negator+SC	Negator+SC	NEGSC