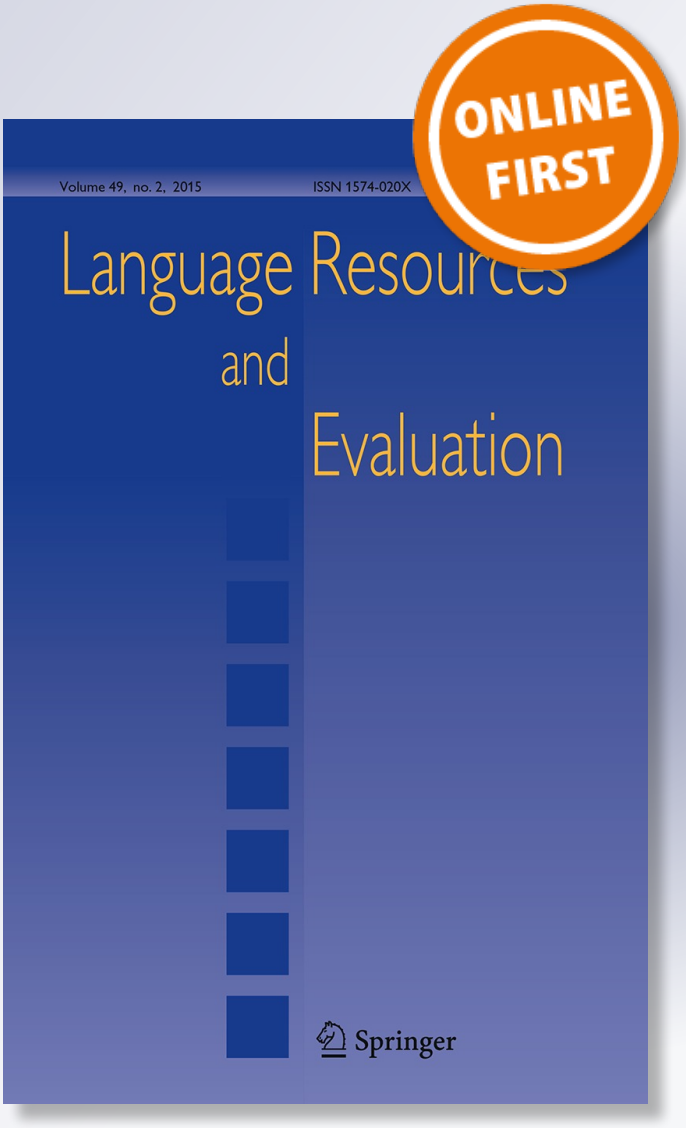
Vietnamese treebank construction and entropy-based error detection

Phuong-Thai Nguyen, Anh-Cuong Le, Tu-Bao Ho & Van-Hiep Nguyen

Language Resources and Evaluation

ISSN 1574-020X

Lang Resources & Evaluation
DOI 10.1007/s10579-015-9308-5



Springer

Lang Resources & Evaluation
DOI 10.1007/s10579-015-9308-5

ORIGINAL PAPER

Vietnamese treebank construction and entropy-based error detection

Phuong-Thai Nguyen[1] · **Anh-Cuong Le**[1] · **Tu-Bao Ho**[2] · **Van-Hiep Nguyen**[3]

Abstract Treebanks, especially the Penn treebank for natural language processing (NLP) in English, play an essential role in both research into and the application of NLP. However, many languages still lack treebanks and building a treebank can be very complicated and difficult. This work has a twofold objective. Firstly, to share our results in constructing a large Vietnamese treebank (VTB) with three levels of annotation including word segmentation, part-of-speech tagging, and syntactic analysis. Major steps in the treebank construction process are described with particular regard to specific Vietnamese properties such as lack of word delimiter and isolation. Those properties make sentences highly syntactically ambiguous, and therefore it is difficult to ensure a high level of agreement among annotators. Various studies of Vietnamese syntax were employed not only to define annotations but also to systematically deal with ambiguities. Annotators were supported by automatic labelling tools, which are based on statistical machine learning methods, for sentence pre-processing and a tree editor for supporting manual annotation. As a result, an annotation agreement of around 90 % was achieved. Our second objective is to present our method for automatically finding errors and inconsistencies in

✉ Phuong-Thai Nguyen
thainp@vnu.edu.vn

Anh-Cuong Le
cuongla@vnu.edu.vn

Tu-Bao Ho
bao@jaist.ac.jp

Van-Hiep Nguyen
nvhseoul@gmail.com

[1] University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam

[2] Japan Advanced Institute of Science and Technology, Nomi, Japan

[3] Institute of Linguistics, Vietnam Academy of Social Sciences, Hanoi, Vietnam

Published online: 27 June 2015

treebank corpora and its application to the construction of the VTB. This method employs the Shannon entropy measure in a manner that the more reduced entropy the more corrected errors in a treebank. The method ranks error candidates by using a scoring function based on conditional entropy. Our experiments showed that this method detected high-error-density subsets of original error candidate sets, and that the corpus entropy was significantly reduced after error correction. The size of these subsets was only about one third of the whole set, while these subsets contained 80–90 % of the total errors. This method can also be applied to languages similar to Vietnamese.

Keywords Treebank · Error detection · Entropy

1 Introduction

Thanks to the development of powerful machine learning methods, natural language processing (NLP) research is currently dominated by corpus-based approaches. Treebanks are used for training word segmenters, part-of-speech taggers, and syntactic parsers, among others. These systems can then be used for applications such as information extraction, machine translation, question answering, and text summarization. The treebanks are also useful for linguistic studies, such as the extraction of lexical-syntactic patterns or the investigation of linguistic phenomena. Treebank construction is a complicated task, and moreover, developing a treebank for a language that has not been the subject of extensive NLP research, such as Vietnamese raises a number of questions concerning the nature of the approach, linguistic issues, and consistency.

Why is linguistic annotation difficult? Linguistic annotation of human languages is difficult because of grammatical complexity and frequently encountered ambiguities. Table 1 shows two examples, one in English part-of-speech tagging (sentences 1–2) and the other in Vietnamese word segmentation (sentences 3–4). In the first example, the word 'can' is an auxiliary in sentence 1, but a noun in sentence 2, and thus there are variations in the way 'can' is tagged. In the second example, the syllable sequence 'sắc đẹp' is a word in sentence 3, but not a word in sentence 4, and thus there are also variations in the way 'sắc đẹp' is segmented. Therefore, building annotated corpora is a costly and labour-intensive task that depends on different levels of annotation such as word segmentation, part-of-speech tagging and syntactic analysis. There are errors even in released data, as shown by the fact that complex data such as treebanks are often released in several versions.[1] In order to speed up annotation and increase the reliability of labelled corpora, various kinds of software tools have been built for format conversion, automatic annotation, and tree

[1] Multi-version treebank publishing has several purposes: error correction, annotation scheme modification, and data addition. For example, major changes in the Penn English Treebank (PTB) Marcus and Marcinkiewicz (1993) upgrade from version I to version II include POS tagging error correction and predicate-argument structure labelling. In the PTB upgrade from version II to version III, more data is appended.

Table 1 Examples of annotation ambiguities

1	I **can** run fast.
2	We drank a **can** of Coke each.
3	Cô ấy$_{she}$ giữ gìn$_{takecare}$ **sắc đẹp**$_{beauty}$. She takes care of her beauty.
4	Bức$_{picture}$ này$_{this}$ màu **sắc**$_{color}$ **đẹp**$_{beautiful}$ hơn$_{more}$. The color of this picture is more beautiful.

editing Pajas and Stepanek (2008). In this paper we have focused on methods for checking errors and inconsistencies in annotated treebanks.

1.1 Previous studies

1.1.1 Treebank construction

The Penn treebank (PTB) for English Marcus and Marcinkiewicz (1993) is the first large syntactically annotated corpus constructed with a good methodology, and good process and evaluation, which results in reliable data. Such treebanks provide rich syntactic information about part of speech, phrase structure, functional and discontinuous constituency (deep structure). Though PTB part-of-speech tagset is less detailed than the tag set of previous POS-tagged corpora such as Brown Corpus and LOB Corpus, due to the recoverability property Marcus and Marcinkiewicz (1993), the end users can convert the PTB tag set into a much richer tag set. Many syntactic parsing studies using various formalisms such as phrase structure grammars Collins (1999), dependency grammars Yamada and Matsumoto (2003), and head-driven phrase structure grammars Miyao and Tsujii (2008) have been carried out successfully using PTB. The PTB phrase structure annotation scheme has been applied to languages such as Korean and Chinese. Treebank development for those languages has contributed to establishing the methodology of PTB.

The Korean treebank (KTB) was developed and evaluated in Han et al. (2002). Korean is an agglutinative language with a very productive inflectional system. POS tags are a combination of a content tag and functional tags. Note that in PTB, only phrasal tags follow this method. Complements and adjuncts are structurally distinguished. If YP is an argument of X, then YP is a sister of X (part (a) in Fig. 1) and If YP is an adjunct of X, then YP is represented as part (b) in Fig. 1. The KTB also uses a number of simple methods to correct POS and constituency errors based on dictionary words and regular expressions.

The Chinese treebank (CTB) Xue et al. (2005) contributes to word segmentation annotation and consistency assurance techniques in the construction of treebanks for an isolating language. For word segmentation, the authors conducted an experiment in manual word segmentation that showed that inter-annotator agreement was not high. However, according to their analyses, much of the disagreement was caused by human error and was not critical. In response, they designed word-hood tests for the word segmentation task. These tests were based on frequency, combination

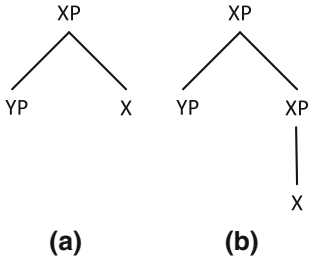


Fig. 1 Distinction between complement and adjunct in Korean treebank

ability, a number of transformations, and the number of syllables. The fact that Chinese words are not marked with tense, case, or gender indicated that there were often two choices of POS criteria: meaning based and distribution based. The authors chose distribution criteria, since it complies with principles in contemporary linguistic theories such as X-bar theory and GB theory.[2] They had pragmatic approaches to quality control and important development phases such as guideline preparation and annotation. For example, in guideline preparation for syntactic bracketing, they tackled ba-construction and bei-construction issues by: (1) studying linguistic literatures, (2) attending Chinese linguistics conferences, (3) conducting discussions with linguistic colleagues, (4) studying and testing their analyses of relevant sentences contained in their corpus, and (5) using special tags to mark crucial elements in these constructions. CTB makes a clearer distinction between constituency and functional tags. Some tags in PTB such as WHNP and WHPP are split in CTB.

There have been a number of published works on Vietnamese word segmentation and POS tagging. These works have often used small, private "home made" corpora. vnQTAG Nguyen et al. (2003), a shared corpus, is one example. This data set, containing 74,756 words, was annotated with word boundaries and POS tags. As with other Vietnamese corpora, there was little description of this corpus. Also, vnQTAG's POS tag set was chosen from a Vietnamese syntactic book. The design of this tag set was based on both meaning and distribution criteria.

Most treebank annotation schemas try to be less specific about linguistic theories. However, two main groups of annotation schemas can be recognized: schemas that annotate the phrase structure as presented above and schemas that annotate the dependency structure. The latter focus on dependency relations between words. Dependency schemes are more suitable for languages with relatively free word order like Czech and Japanese, since grammatical functions can be indicted without lots of indications of movement. Recently, Rambow (2010) have had an excellent discussion about dependency representations and phrase structure representations for syntax.

[2] This choice emphasizes the similarity between Chinese and other languages.

1.1.2 Treebank error detection

Dickinson and Meurers (2003) proposed three techniques to detect part-of-speech tagging errors. The main idea of their first technique was to consider variation n-grams, which occur more than once in the corpus and include at least one difference in their annotation. For example, "centennial year" is a variation bi-gram which occurs in the Wall Street Journal (WSJ), a part of Penn treebank corpus Marcus and Marcinkiewicz (1993) with two possible tagging[3] "centennial/JJ year/NN" and "centennial/NN year/NN". Of these, the second tagging is correct. Dickinson found that a large percentage of variation n-grams in WSJ have at least one instance (occurrence) of an incorrect label. However, using this variation n-gram method, linguists have to check all instances of variation n-grams to find errors. The other two techniques take into account more linguistic information including tagging-guide patterns and functional words.

Dickinson (2006) presented an error correction method employing off-the-shelf POS taggers.[4] The method includes three steps: firstly, training the tagger on the entire corpus; secondly, running the trained tagger over the same corpus; thirdly, for the positions the variation ngram detection method Dickinson and Meurers (2003) flags as potentially erroneous, choosing the label output by the tagger. Dickinson's paper also presented a treebank transformation method to improve POS tagging accuracy, which resulted in improvements in error correction. The method converts original POS tags into ambiguity tags in order to reduce ambiguity in the original data. Treebank transformation techniques have been used for both POS tagging, as mentioned in Dickinson's paper, and syntactic parsing Johnson (1998), Klein and Manning (2003). Treebank transformation is often carried out as a preprocessing step for different tagging and parsing methods.

Dickinson (2008) reported a method to detect ad-hoc treebank structures. He used a number of linguistically-motivated heuristics to group context-free grammar (CFG) rules into equivalent classes by comparing the right hand side (RHS) of rules. For example, one heuristic suggests that CFG rules of the same category should have the same head tag and similar modifiers, but can differ in the number of modifiers they have. By applying these heuristics, the RHS sequences[5] ADVP RB ADVP and ADVP, RB ADVP can be grouped into the same class. Classes with only one rule, or rules which do not belong to any class are problematic. Dickinson evaluated the proposed method to analyse several types of errors in the Penn treebank Marcus and Marcinkiewicz (1993). However, in a similar way to Dickinson and Meurers (2003), this study proposed a method to determine candidates of problematic patterns (ad hoc CFG rules instead of variation n-grams) but not problematic instances of those patterns.

Yates et al. (2006) produced a study on detecting parser errors using semantic filters. Firstly, the syntactic trees—the output of a parser—are converted into an

[3] JJ: adjective, NN: noun

[4] Note that before Dickinson, Halteren (2000) pointed out that POS taggers can be used to enforce consistency.

[5] ADVP: adverbial phrase, RB: adverb

intermediate representation known as relational conjunction (RC). Then, using the Web as a corpus, RCs are checked using various techniques including point-wise mutual information, verb sampling tests, text-runner filters, and question answering (QA) filters. For evaluation, error rate reductions of 20 and 67 % were reported when tested on the PTB and TREC, respectively. The interesting point of their paper was that information from the Web was utilized to check for errors.

Novak and Razimova (2009) used the association rule mining algorithm Apriori to find annotation rules, and then to search for violations of these rules in corpora. They found that violations are often annotation errors. They reported an evaluation of this technique performed on the Prague Dependency Treebank 2.0, presenting an error analysis which showed that in the first 100 detected nodes, 20 contained an annotation error. However, this was not an intensive evaluation.

1.2 A summary of our work

1.2.1 Vietnamese treebank Construction

There are a number of important characteristics of the Vietnamese language that impact greatly on the treebank construction. First, the smallest unit in the formation of Vietnamese words is the syllable. Words can have just one syllable (for example '$\text{đẹp}_{beautiful}$') or be a compound of two or more syllables (for example 'màu sắc_{color}'). Like many other Asian languages such as Chinese, Japanese and Thai, there is no word delimiter in Vietnamese. The space is a syllable delimiter but not a word delimiter, so a Vietnamese sentence can often be segmented in many ways. Second, Vietnamese is an isolating language in which words do not change their forms according to their grammatical function in a sentence. Table 2 shows an example. Vietnamese words 'anh ấy_{he}' and 'ra_{come}' function as the subject and the main verb respectively in sentence 1, while they function as the complements of 'bảo_{ask}' in sentence 2. However, in both sentences, these words do not change their forms, while English translation sentences 1e-2e require different word forms ('he'-'him' and 'comes'-'to come'). Third, the Vietnamese syntax conforms to the subject-verb-object (SVO) word order as illustrated in examples we considered so far (Tables 1, 2).

Since Vietnamese has a relatively restrictive word order and often relies on the order of constituents to convey important grammatical information, we chose to use constituency representation of syntactic structures. For languages with a freer word order such as Japanese or Czech, dependency representation is more suitable. We applied the annotation scheme proposed by Marcus et al. Marcus and

Table 2 An example about isolating property of the Vietnamese language

1	$\textbf{Anh ấy}_{he}$ $\textbf{ra}_{come}$ Hà Nội_{Hanoi}.
1e	**He comes** to Hanoi.
2	$\text{Giám đốc}_{manager}$ bảo_{ask} $\textbf{anh ấy}_{him}$ $\textbf{ra}_{come}$ Hà Nội_{Hanoi}.
2e	The manager asks **him to come** to Hanoi.

Marcinkiewicz (1993). This approach has been successfully applied to a number of languages such as English, Chinese, and Arabic. For Vietnamese, there are three annotation levels including word segmentation, POS tagging, and syntactic labeling. Our main goal was to build a corpus of 70,000 word segmented sentences, 20,000 POS tagged sentences, and 10,000 syntactic trees.[6] Treebank construction is a very complicated task in which the major phases include investigation, guideline preparation, tool building, raw text collection, and annotation. Actually this is an iterative process involving three phases: annotation, guideline revision, and tool upgrade. We drew our raw texts from the news domain, with the Youth (Tuổi Trẻ), an online daily newspaper, focusing on social and political topics, as our source.

In order to deal with ambiguities occurring at various levels of annotation, we systematically applied linguistics analysis tests such as deletion, insertion, substitution, questioning, and transformation Nguyen (2009). Notions for these techniques were described in the guideline documents with examples, arguments and alternatives. These techniques originated in the literature or were proposed by members of our group. For automatic labeling tools, we used advanced machine learning methods such as conditional random fields (CRFs) for POS tagging or lexicalized probabilistic context-free grammars (LPCFGs) for syntactic parsing. These tools helped us speed up the annotation process. We also used a tree editor to support manual annotation.

Our treebank project is a branch project of a national project which aims to develop basic resources and tools for Vietnamese language and speech processing (VLSP). In addition to a treebank, the VLSP project also develops other text-processing resources and tools including a Vietnamese machine readable dictionary, an English-Vietnamese parallel corpus, a word segmenter, a POS tagger, a chunker, and a syntactic parser. During the annotation process, tools are trained using treebank data, and then are used to support treebank construction as a preprocessing step.

After finishing the treebank project, we achieved our goal in terms of corpus size, annotation agreement, and usability for text-processing tools. Since 2010, the Vietnamese treebank (VTB) and other resources and tools developed by the VLSP project have been shared on the VLSP web page.[7] Sections 2.5 and 2.6 will give more analysis about the treebank status.

1.2.2 Treebank error detection

In this paper, we introduce a learning method based on conditional entropy for detecting errors in treebanks. Our method, using ranking, can detect erroneous instances of variation ngrams[8] in treebank data (Fig. 2). This method is based on the entropy of labels, given their contexts. Our experiments showed that conditional

[6] Steedman et al. (2003) showed that a training set size of around 10,000 syntactic trees was good for English parsing since when using a larger training set, improvement in parsing performance was small (as tested on Collins' parser).

[7] http://vlsp.vietlp.org:8080/demo/

[8] This term has the same meaning as the term 'variation nuclei' in Dickinson and Meurers (2003). In our paper, a variation n-gram is an n-gram which varies in how it is labelled because of ambiguity or annotation error. Contextual information, such as surrounding words, is not included in an n-gram.

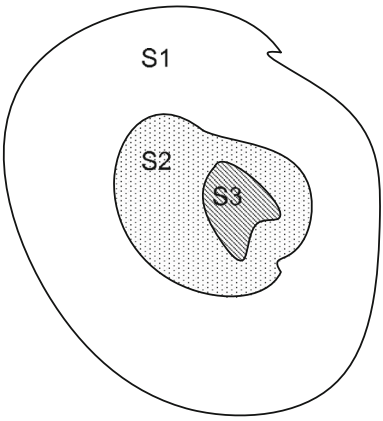


Fig. 2 Conceptual sets. *S1* The whole treebank data; *S2* data set of variation ngrams; *S3* error set (supposed to be the region with highest entropy)

entropy was reduced after error correction, and that by using ranking, the number of checked instances could be reduced drastically. We used Vietnamese treebank Nguyen et al. (2009) for the experiments.

Our method inherits the idea of variation ngram/nuclei from the work of Dickinson and Meurers (2003), although it improves the capability of detecting erroneous instances. Our work differs from Dickinson (2006) in that we do not require an available POS tagger. Instead, we sort error candidates, employ entropy for error detection, and experiments on not only POS tagged data, but also word-segmented data sets that show the effectiveness of the entropy-based method.

1.3 Organization of the paper

The rest of this paper is organized as follows. In Sect. 2, we present the main aspects of Vietnamese treebank construction including annotation schemes, guideline preparation for three annotation levels, tools, annotation process, and preliminary results on treebank and tool distribution. In Sect. 3 we present a mathematical relationship between entropy and annotation errors, an entropy-based error detection method, and experimental results for error detection with discussion. Finally, conclusions are drawn, and future work is proposed in Sect. 4.

In this paper, Vietnamese examples are annotated with English words as subscripts, except for proper nouns and numbers. Since Vietnamese is an isolating language, English subscripts are often in base form. There are several special subscripts expressing grammatical information including the 'past', 'continuous', 'future', and 'passive' tenses. In the reference section, there are selected Vietnamese books and journal papers in which only two are in English[9] Nguyen (2009); Thompson 1987), and the others are in Vietnamese.

[9] Online versions at: http://ir.library.osaka-u.ac.jp/metadb/up/LIBRIWLK01/riwl_001_019.pdf; Ćhttp://www.sealang.net/archives/mks/THOMPSONLaurenceC.htm

2 Vietnamese treebank construction

2.1 Word segmentation

2.1.1 Word types

With regard to their structure, Vietnamese words can be divided into a number of types including single-syllable words, coordinated compound words, subordinated compound words, reduplicative words, and accidental compound words. As shown in Table 3, single-syllable words only cover a small proportion while two-syllable words account for the largest proportion of the whole vocabulary. Forming that vocabulary is a set of 7729 syllables, higher than the number of single words. The syllables which are not single words are bound morphemes,[10] which can only be used as part of a word but not as a word on its own. The coordinated compound words, specific to Vietnamese, are words in which their parts—each part can be a word, single or compound words—are parallel in the sense that their meanings are similar and their order can be reversed. The meaning of a coordinated compound is often more abstract than the meanings of its parts. The proportion of this kind of words is about 10 % of the number of compound words according to the statistics in the Vietlex dictionary. Reduplicative words (such as 'đất đai_{land}', 'làm lụng_{work}') are compounds whose parts have a phonetic relationship. This kind of words is specific to Vietnamese, although their proportion is small. The identification of reduplicative words is normally deterministic and not ambiguous. Accidental compounds are non-syntactic compounds containing at least two meaningless syllables such as '$\text{đười ươi}_{orangutan}$', 'bù nhìn_{puppet}'. Subordinated compound words (SCWs) are the most problematic. A SCW can be considered as having two parts, a head and a modifier. Normally, the head goes first and then the modifiers. SCWs make up the largest proportion in the Vietnamese dictionary. Generally, discrimination between SCW and phrase is problematic because SCW's (syntactic) structure is similar to that of a phrase. This is a classical but persistent problem in Vietnamese linguistics.

In addition to the word types mentioned above, we consider the following types in the word segmentation phase: idioms, proper names, date/time and number expressions, foreign words, and abbreviations. Note that sentences are segmented into word sequences in which words are not labeled with type information. However, in our annotation guidelines, word segmentation rules are organized following word types.

2.1.2 Word definition

There are many approaches to word definition such as those based on morphology, syntax, meaning or linguistic comparison. Since Vietnamese words are not marked with respect to number, case or tense, the morphology-based approach is not very applicable. We mostly rely on an approach based on the syntactic role and

[10] They may have a meaning ('trường_{long}', 'hàn_{cold}') or not ('lẽo', 'nhánh')

Table 3 Word length statistics from a popular Vietnamese dictionary, made by the Vietnam Lexicography Center (Vietlex)

Length	Words	Percentage
1	6303	15.69
2	28,416	70.72
3	2259	5.62
4	2784	6.93
5	419	1.04
Total	40,181	100

combination ability of words, so that we consider words to be syntactic atoms Sciullo and Williams (1987) in the sense that it is impossible to analyze the word structure using syntactic rules (except subordinated compounds), or that words are the smallest unit which is syntactically independent. We do not use meaning as a word definition, but we make use of the non-compositionality property of a large proportion of compound words.

From the application point of view, the word definition should support applications as much as possible. For example, machine translation researchers may prefer a good match between Vietnamese vocabulary and foreign languages' vocabulary. The problem is that there are so many foreign languages which are different in terms of linguistic properties and word characteristics. Lexicographers (dictionary makers) may want to extract candidates of collocations and new words from texts, which need to have their meaning explained. For such applications, syntactic parsers can be used since they can identify and extract phrases. The application considerations are important. However at this stage of the resource development of Vietnamese NLP, we have concentrated on word segmentation for other fundamental tasks such as POS tagging, chunking, syntactic parsing than about other applications.

2.1.3 Word segmentation guidelines

In the annotation phase, we used dictionaries as a reference. In fact, dictionary words can be considered to be candidates for word segmentation and the right segmentation will be chosen based on context. This is not a very difficult task for humans. We also applied techniques to identify new (compound) words. For repeated words, there are linguistic rules Nguyen (2004) which well-trained annotators can apply without much difficulty. For coordinated and subordinated compound words, we used word-hood tests which have been discussed in various Vietnamese linguistic studies:

Tests for word-hood verification (without loss of generality, considering a sequence of two syllables AB):

- Stress: in pronunciation of AB, if A or B is stressed while the other is not, then AB is likely to be a word.
- Bound morpheme: if A or B (or both) is a bound morpheme, then AB is likely to be a word.
- Order: if AB is inverse to the common head-first-then-modifier order in Vietnamese syntax, then AB is likely to be a word.
- Non-compositionality: if the meaning of AB can not be inferred from the meanings of A and B, then AB is likely to be a word.
- Parallel: if the meaning of A and the meaning of B are similar, and A and B can be reordered, then AB is likely to be a coordinated compound word.
- Transformation 1 (insertion): if we can insert C, C', ... between A and B, then AB is not likely to be a word. The more productive the transformation is, the less likely to be a word AB is.
- Transformation 2 (substitution): if A (or B) can be substituted by A', A", ... (or B', B", ...) of the same type, then AB is not likely to be a word. The more productive the transformation is, the less likely that AB is a word.

In fact, our word segmentation guidelines are much more specific than the previous list of tests. However, as shown by a prior study of Chinese treebank construction Xue et al. (2005), the specification of such general word-hood tests can help annotators systematically understand word identification criteria, and therefore improve the inter-annotator word segmentation agreement. These word-hood tests can be used directly or indirectly in case there are more specific tests (guidelines) of the same type.

In practice, in verifying whether a syllable compound is a word or not, annotators often have to use multiple tests. The satisfaction of one test reflects only one aspect —phonetic, structural, and syntactic transformation possibilities—of a word. There exist compounds that satisfy all or most of the tests such as 'châu chấu$_{grasshopper}$', 'tạp chí$_{magazine}$' (words). There are also sequences that do not satisfy any test, such as 'uống$_{drink}$ nước$_{water}$', 'ăn$_{eat}$ xôi$_{stickyrice}$' (phrases). Between these opposite poles, there are sequences that satisfy one, two, or several tests only. Such sequences, which are often SCWs, form a source of inconsistency. We try to maintain the consistency of annotation as much as possible. For example, if 'đàn anh$_{a\ male\ senior}$' is identified as a word, then by substitution transformation, 'đàn chị$_{a\ female\ senior}$' and 'đàn em$_{a\ junior}$' should be recognized too. Note that in this example, the transformation can be considered not productive since it results in two possible compounds only.

2.2 Part-of-speech tagging

2.2.1 Part-of-speech tag set and annotation guidelines

In Vietnamese syntactic studies, there are two common approaches to classifying words into POSs. The first approach is based on the combination ability and syntactic functions of words (or in other words, distribution), while the other relies on word meaning. In fact, these approaches are often combined Diep (2005). We

choose the first view, combination ability and syntactic function, for our POS tag set design since words with different meanings can have the same syntactic function. Therefore our POS tags do not contain semantic information. Note that in NLP, POS tagging studies often make use of local lexical information such as surrounding words and POSs, rather than use phrase-structure information. In practice the pipeline processing, or incremental approach, is quite popular when building sentence analysis systems.[11] For example, in order to parse a sentence, the necessary processing steps include word segmentation, POS tagging, and syntactic parsing. The process operates sequentially with the output of one step providing the input to the next step. Our POS tags do not contain sub-categorization information (e.g. transitive/intransitive verbs, verbs followed by clauses, etc.). Where "extra" information such as semantic and sub-categorization is necessary, higher levels of analyses such as word sense disambiguation and syntactic parsing are required. From these reasons, we choose a medium level of tag details. Later, we will discuss the refinable property of a number of tags.

Table 4 shows our POS tag set. Vietnamese parts of speech do not necessary correspond directly with English parts of speech of the same name. The class of adverbs in Vietnamese is a closed class (or a class of function words), while in English the class of adverbs is an open class (or a class of content words). Vietnamese adverbs express time (such as 'đã_{past}', '$\text{đang}_{continuous}$'), degree (such as 'rất_{very}', 'hơi_{rather}'), and negation (such as 'không_{not}'). Therefore the number of adverbs in Vietnamese is much smaller than that in English. Other words that change or qualify the meaning of verbs are classified in Vietnamese as adjectives.[12]

There are current controversies about how some Vietnamese words should be tagged with POS. Classifier words such as 'cái', 'con' are examples. Vietnamese countable nouns often must be preceded by these classifiers when these nouns are being counted or specified (e.g. '*cái* bàn_{table}', '*con* $\text{gà}_{chicken}$'). Some argue that this group of words should be considered an independent part of speech or a sub class of noun. Recent studies have showed that these words can be considered as (classifier) nouns Cao (2007). More specifically, these words can serve as the head of a noun phrase. Another example concern the verb and adjective parts of speech. Many linguists agree that we can use *lexical evidences* such as 'đã_{past}', '$\text{đang}_{continuous}$', 'sẽ_{future}', ... to identify verbs, 'rất_{very}', 'hơi_{rather}', '$\text{quá}_{extremely}$', 'lắm_{so}', ... to identify adjectives. However, Cao (2007) showed that many verbs, such as emotional ones, can co-occur with 'rất_{very}', 'hơi_{rather}', '$\text{quá}_{extremely}$', 'lắm_{so}', ..., while hundreds of adjectives (such as extreme adjectives) can not occur concurrently with that group of adverbs. Cao states that we should merge verb and adjective parts of speech as the predicative part of speech. Although we do not follow his opinion, his argument helps us understand some of the limitations of the *lexical evidence* techniques we used.

In the POS annotation guidelines, we list ambiguous cases and describe tests and examples for POS disambiguation. For example, the set of directional words 'ra_{out}', 'vào_{in}', 'lên_{up}', 'xuống_{down}', ... are ambiguous between verb, preposition, and

[11] The other approach is joint processing, in which all tasks are carried out simultaneously.

[12] This classification is widely accepted in the Vietnamese linguistic community.

Table 4 Vietnamese treebank POS tag set

No	POS tag	Description	Example
1	N	Noun	tiếng$_{language}$, nước$_{country}$, thủ đô$_{capital}$
2	Np	Proper noun	Nguyễn Du, Việt Nam, Bill Gates
3	Nc	Classifier noun	con, cái, đứa, bức
4	Nu	Unit noun	mét$_{meter}$, cân$_{kilo}$, giờ$_{hour}$, đồng$_{pound}$
5	V	Verb	ngủ$_{sleep}$, ngồi$_{sit}$, đọc$_{read}$, thích$_{like}$
6	A	Adjective	tốt$_{good}$, xấu$_{bad}$, cao$_{high}$, thấp$_{short}$
7	P	Pronoun	tôi$_{I,me}$, chúng tôi$_{we,us}$, hắn$_{he,him}$
8	L	Determiner	mỗi, từng$_{each}$, mọi$_{every}$,các, những, mấy
9	M	Number	mười$_{ten}$, dăm$_{aroundfive}$, vài$_{several}$
10	R	Adverb	đã$_{-ed}$, sẽ$_{will}$, đang$_{-ing}$, vừa$_{just}$, rất$_{very}$
11	E	Preposition (subordinating conjunction)	trên$_{on}$, dưới$_{under}$, trong$_{int}$, ngoài$_{out}$
12	C	Coordinating conjunction	và$_{and}$, với$_{each}$, cùng, vì vậy , tuy nhiên, ngược lại
13	I	Interjection	ôi$_{oh}$, chao$_{wow}$, a ha
14	T	Particle	à, a, ạ, chăng, chứ (modal particle)
15	B	Borrowed/foreign word	Internet, email, video, chat
16	Y	Abbreviation	OPEC, WTO, HIV
17	X	Can-not-classified word	

adverb POSs. For instance, 'ra' is a verb in 'Tôi_{I} ra_{go} Hà Nội' (insertion of 'đã$_{past}$', 'đang$_{continuous}$', 'sẽ$_{future}$', …), an adverb in 'Tôi$_{I}$ nghĩ$_{find}$ra$_{out}$ giải pháp$_{solution}$' (showing result), and a preposition in 'Tôi$_{I}$ đi$_{go}$ ra_{to} Hà Nội' (have a noun complement).

2.2.2 *Refinable properties*

Based on lexical information, a number of tags such as pronoun, adverb, conjunction, and particle can be easily—with less ambiguity—split into more specific sub-tags. This is similar to the recoverable[13] property mentioned by Marcus et al. Marcus and Marcinkiewicz (1993). For example, the pronoun tag P can be split into the vocative pronoun (such as 'tôi$_{I,me}$', 'chúng tôi$_{we,us}$'), the deterministic pronoun (such as 'đây$_{this}$', 'đó$_{that}$'), and the interrogative pronoun (such as 'ai$_{who}$', 'gì$_{what}$'). Another example is the adverb tag R. This tag can be split into three subtypes reflecting the relative position of an adverb in a sentence including beginning of the sentence (such as 'thỉnh thoảng$_{somtimes}$', 'bỗng dưng$_{suddenly}$'), preceding the modified verb (such as 'cũng$_{also}$', 'sẽ$_{future}$') or following the verb (such as 'rồi$_{already}$', 'nữa$_{again}$'). This property can be useful in cases in which linguists want to investigate the behaviour of words belonging to these subtypes.

In designing this tag set, we did not make distinctions within the syntactic structure. For example, we do not distinguish noun-modifier adjectives from predicative adjectives and verb-modifier adjectives and vice versa. Such distinctions can be made from the information about the adjective's position in the parse tree (e.

[13] This term came from the fact that the design for the Penn Treebank tag set was based on the simplification of the Brown Corpus tag set.

g. if the parent node's tag is NP, then the adjective is a noun modifier) in the parsed version of the corpus.

2.3 Syntactic annotation

2.3.1 Syntactic tag set

Our syntactic tag set contains three types of tags including constituency (Table 5), function (Table 6), and null element (Table 7). The design of constituency tags is less controversial than that of the POS tag set. Each phrase tag XP often has a corresponding POS tag X, as the POS of the XP's head is X. Tags with a WH prefix such as WHNP, WHAP are used for labelling phrases containing the interrogative words, used in question sentences. Another design option is to represent WH as a functional tag as in the Chinese treebank. There are several clausal tags representing statement sentences, question sentences, and also subordinate clauses. Each functional tag represents a specific kind of complement or adjunct. Considering that head identification is important, we use the tag H to label phrases' head. If a phrase has more than one head connected by coordination conjunctions or commas, then all heads will be labelled with the H tag. Since other treebanks such as PTB and CTB often do not use head tag, researchers in syntactic parsing such as Collins Collins (1999), Klein and Manning (2003) use heuristic rules to determine the head of CFG rules. Machine learning methods such as the expectation maximization (EM) algorithm also can be used to recover the 'hidden' head in this case Chiang and Bikel (2002). Null element tags are often used for representing deep structures of adjective clauses, ellipsis, passive voice, and topic.

Table 5 Vietnamese treebank constituency tags

No	Constituency tag	Description
1	NP	Noun phrase
2	VP	Verb phrase
3	AP	Adjective phrase
4	RP	Adverb phrase
5	PP	Prepositional phrase
6	QP	Quantitative phrase
7	MDP	Modal phrase
8	UCP	Coordinated phrase in which components are not the same type
9	LST	List mark phrase
10	WHNP	Interrogative noun phrase ('ai$_{who}$', 'cái gì$_{what}$', 'con gì$_{which}$')
11	WHAP	Interrogative adjective phrase ('lạnh$_{cold}$ thế nào$_{how}$', 'đẹp$_{beautiful}$ ra sao$_{how}$')
12	WHRP	Interrogative adverb phrase
13	WHPP	Interrogative prepositional phrase ('với$_{with}$ ai$_{whom}$', 'bằng$_{by}$ cách$_{method}$ nào$_{which}$')
14	S	Statement sentence
15	SQ	Question sentence
16	SBAR	Subordinate clause (modifying noun, verb, and adjective)

Table 6 Vietnamese treebank functional tags

No.	Functional tag	Description
1	H	Head of phrase
2	SUB	Subject
3	DOB	Direct object
4	IOB	Indirect object
5	TPC	Topic
6	PRD	Predicate
7	LGS	Logical subject
8	EXT	Frequency or range complement
9	VOC	Vocative
10	TMP	Temporal adjunct
11	LOC	Location adjunct
12	DIR	Direction adjunct
13	MNR	Manner adjunct
14	PRP	Purpose adjunct
15	CND	Condition adjunct
16	CNC	Cnc adjunct
17	ADV	Adverbial adjunct
18	EXC	Exclamation sentence
19	CMD	Command sentence

Table 7 Vietnamese treebank null-element tags

No.	Null-element tag	Description
1	*T*	Null element (trace within sentence)
2	*E*	Null element in ellipsis phenomenon
3	*0*	Null element in complementizer

2.3.2 Sentence and phrase analysis techniques

In Vietnamese, words belonging to a number of classes such as verb, adjective, noun and preposition can be the predicate of a sentence. Additionally, the isolating property makes Vietnamese sentences more structurally ambiguous. A kind of 'morphological' property of Vietnamese is that there are functional words expressing number (before nouns), tense (before verbs), capability (before adjectives), etc. However, the use of these words is often not a morpho-syntactic constraint, and speakers or writers just use those functional words when they are important for expressing the meaning of a sentence (i.e. to avoid misunderstanding). Therefore, in sentence analysis, the test of insertion of functional words is important.

The annotation of real texts relies on various techniques because ambiguity may occur in any step of phrase structure analysis, such as determining the head element, discriminating between possible syntactic patterns (especially subcategorization frames), and discriminating between complements and adjuncts. Important sentence analysis techniques include deletion, substitution, insertion, transformation, and question formation. These techniques exploit combination ability, word order, and functional words in order to disambiguate between possible structures. Table 8 shows some examples of such techniques.

As shown in the examples above, to identify linguistic units (e.g. an adverbial phrase), syntactic behaviors (e.g. reorderable) can be verified by transformation. If a transformed sentence is correct, then the corresponding analysis is chosen. Otherwise, it is rejected. In practice, the incorrectness of a transformed sentence can be caused by a reason on the syntactic level, the semantic level, or the pragmatic level. It is not always easy to separate these levels.

2.3.3 Existential and passive sentences

The subject identification of existential sentences like 'Trên$_{on}$ bàn$_{table}$ đặt$_{put}$ một$_{a}$ lọ$_{vase}$ hoa$_{flower}$.' or 'Nhà$_{house}$ đang$_{continuous}$ xây$_{build}$..' is quite controversial. Such sentences can be composed mechanically by omitting the first argument (e.g. agent) of the predicate, and moving one of the other arguments (e.g. recipient) toward the beginning of the sentence. The commonality of existential sentences is the consequence of the topic sensitive property of Vietnamese language. Recent syntactic studies Nguyen (2009) showed that 'trên$_{on}$ bàn$_{table}$' or 'nhà$_{house}$' can be considered as the subject of such sentences, while a number of previous studies recognized 'trên$_{on}$ bàn$_{table}$' as an adverbial phrase, or 'nhà$_{house}$' as a moved object, and there is no subject. To ensure the consistency of our subject-predicate approach, we label those phrases functionally as subject. We use null element tags with trace indices to imply that logically, such phrases are the moved argument of the predicate.

Table 8 Examples of disambiguation tests

Ambiguity	Test	Example
S vs NP VP	Insertion of 'đã', 'đang', 'sẽ', ...	Giám đốc$_{manager}$ bảo$_{ask}$ anh ấy$_{him}$ ra$_{come}$ Hà Nội$_{Hanoi}$ *Giám đốc$_{manager}$ bảo$_{ask}$ anh ấy$_{him}$ **đang**$_{continuous}$ ra$_{come}$ Hà Nội$_{Hanoi}$ The manager asks him to come to Hanoi.
'là' NP vs 'là' S	Insertion of 'mà'	Tôi$_{I}$ là$_{am}$ người$_{man}$ đặt cược$_{bet}$ ít$_{little}$ nhất$_{most}$. Tôi$_{I}$ là$_{am}$ người$_{man}$ **mà**$_{who}$ đặt cược$_{bet}$ ít$_{little}$ nhất$_{most}$. I am the man who bets the smallest amount.
PRP labeling	Insertion of 'để'	Nó$_{He}$ đi làm$_{work}$ kiếm$_{earn}$ tiền$_{money}$ nuôi$_{support}$ em$_{brother}$. Nó$_{He}$ đi làm$_{work}$ **để**$_{for}$ kiếm$_{earn}$ tiền$_{money}$ nuôi$_{support}$ em$_{brother}$. He works for earning money to support his brother.
Adverbial identification	Permutation (reordering)	**Ở**$_{In}$ **quê**$_{hometown}$, mẹ$_{mother}$ đang$_{continuous}$ cấy$_{plant}$ lúa$_{rice}$. Mẹ$_{mother}$ **ở**$_{in}$ **quê**$_{hometown}$ đang$_{continuous}$ cấy$_{plant}$ lúa$_{rice}$. Mẹ$_{mother}$ đang$_{continuous}$ cấy$_{plant}$ lúa$_{rice}$ **ở**$_{in}$ **quê**$_{hometown}$. In hometown, the mother is planting rice.

The translation of sentences like 'Nhà$_{house}$ đang$_{continuous}$ xây$_{build}$.' into English normally results in a passive sentence form. However, in Vietnamese the syntactic structure of such sentences is often not considered as a passive form although 'nhà$_{house}$' logically is the object of 'xây$_{build}$'. So it is simple to create sentences with objects as subjects in Vietnamese. A number of researchers studying passive sentence form in Vietnamese have argued that such sentences must contain 'bị' or 'được' as a predicate. Vietnamese speakers use 'bị', 'được' when they are necessary to express a 'passive' meaning. The role of passive sentence form in Vietnamese syntax is often downplayed, and it is even controversial whether such sentence forms should be recognized in Vietnamese.

2.3.4 Linguistic issues

Treebank construction is a good place for the application of linguistic theories. However, there are still disagreements among linguists about many linguistic issues in the Vietnamese language. In previous sub-sections, we have shown several examples of such disagreements. When consensus among linguists is not available, we choose linguistic solutions that are least controversial, well established, or more appropriate to our treebank construction approach. For example, in the early 1990s there were debates over whether Vietnamese sentence structure is topic prominent Cao (2007) or subject prominent, as it is usually classified. We chose the more conventional view. Similarly, our treebank relies more on the subject-predicate structure.

We believe that the application of several contemporary linguistic theories, such as government and binding theory and head-driven phrase structure grammars to Vietnamese is new Nguyen (2009). However, since other treebanks Xue et al. (2005), Han et al. (2002) have claimed the influence, but not the domination, of these theories, we can point out here some ideas about various studies on Vietnamese syntax which explicitly or implicitly agree. First, the head of a phrase characterizes the syntactic properties of that phrase. The head is the only element which can have syntactic relations outside that phrase. Second, sub-categorization information including what kind of complements and their orders is determined by the predicate word (or lexical head). These are suitable for some principles of GB theory such as head principle, and projection principles.

2.4 Tools

We have designed a tool (Fig. 3) to support annotators in almost all phases of the annotation process. Our editor provides the following functions:

- Editing and viewing trees in both text mode and graphical mode,
- Viewing log files, highlighting modifications,
- Searching by words or syntactic patterns,
- Predicting errors (edit, spell, or syntax),
- Computing annotation agreement and highlighting differences,
- Computing several kinds of statistics.

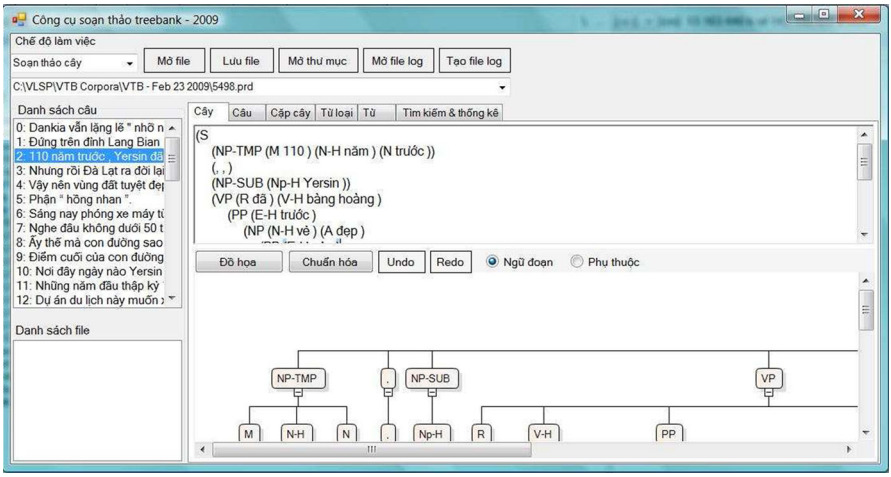


Fig. 3 Vietnamese treebank editor

In support of the treebank construction, we have developed an exchange format we call vnSynAF, that provides a syntactic annotation framework which conforms to the ISO standard framework SynAF. The SynAF framework is built on top of an XML-based annotation scheme that ISO recommends for encoding treebanks. Our tool also supports the bracketing representation (or Lisp style) of the Penn English Treebank. These formats have equivalent abilities, so software can provide translation between the two. For word segmentation, we used vnTokenizer, a highly accurate segmenter which uses a hybrid approach to automatically tokenize Vietnamese text. This approach combines a finite-state automata technique, regular expression parsing, and a maximal-matching strategy augmented by statistical methods that resolve ambiguities of segmentation Phuong et al. (2008). We also used JVnTagger, a POS tagger based on Conditional Random Fields Lafferty et al. (2001) and Maximum Entropy Berger et al. (1996). This tagger was also developed in the framework of the VLSP project. The training data size is 10,000 sentences. Experiments with 5-fold cross validation showed that F1 scores for CRFs and Maxent were 90.40 and 91.03 %, respectively. Another tool we used was a syntactic parser based on Lexicalized Probabilistic Context-free Grammars (LPCFGs). Researchers in another VLSP group have customized Bikel's parser[14] for parsing Vietnamese text. This parser is well designed and easy to adapt to new languages. The same group implemented a Vietnamese language package which handles treebanks, training, and finding the head of CFG rules and word features. This parser can output text with constituent tags with or without functional tags.

[14] http://www.cis.upenn.edu/dbikel/software.html

2.5 Annotation process and agreement

Because the word segmentation tool was available before the start of our project, it was immediately adopted for the first annotation level (word segmentation). As for the other annotation levels (POS tagging and syntactic parsing), several thousand sentences were labelled manually. Then a POS tagger and a parser were trained on a bimonthly basis, so that the annotation task became semi-automatic. Our annotation process requires that each sentence was annotated and revised by at least two annotators. The first annotator labeled raw sentences or revised automatically analyzed sentences. Then the second annotator revised the output of the first annotator. In addition, we checked the corpus for syntactic phenomena. For example, we checked direction words and questions with the support of available tools. Thus, there were many sentences that were revised more than once.

Annotation agreement measures the similarity between the annotations created by two independent annotators. Because this problem is similar to the parsing evaluation, we use the parseval measure Black et al. (1991). First, syntactic constituents in the form (i, j, label) are extracted from syntactic trees. Then the tree comparison problem is transformed into constituent comparison. We compute three kinds of measurement: constituent and function similarity, constituent similarity, and bracket similarity. By using this method, we can evaluate both overall and constituency agreements. The annotation agreement A between two annotators can be computed as follows:

$$A = \frac{2 \times C}{C_1 + C_2} \tag{1}$$

where C_1 is the number of constituents in the first annotator's data set, C_2 is the number of constituents in the second annotator's data set, and C is the number of identical constituents. For example, considering two possible trees of the sentence 'Tôi$_I$ đi$_{come}$ Nha Trang$_{NhaTrang}$ dự$_{attend}$ hội thảo$_{conference}$' as follows:

```
(S (NP Tôi)
   (VP đi
       (NP Nha Trang))
   (VP dự
       (NP hội thảo)))

(S (NP Tôi)
   (VP đi
       (NP Nha Trang)
       (VP dự
           (NP hội thảo))))
```

$$C_1 = 6; C_2 = 6; C = 5; A = 10/12 = 0.83.$$

We carried out an experiment involving 3 annotators (represented as A_1, A_2, and A_3 in Table 9). They annotated a set of 100 randomly-selected sentences. Table 9

Table 9 Annotation agreement

Test	A_1-A_2 (%)	A_2-A_3 (%)	A_3-A_1 (%)
Full tag	90.32	91.26	90.71
Constituent tag	92.40	93.57	91.92
Bracketed only	95.24	96.33	95.48

shows that we achieved a full-tag agreement (bracketing, constituency, and functional labels are correct) around 90 %. According to previous studies such as Xue et al. (2005), this level of inter-annotator agreement is acceptable. There was a number of major kinds of disagreements. The first was disagreements caused by XP-attachment ambiguities. For example, for a prepositional phrase, it might be ambiguous between possible attachments to preceding nouns, preceding verbs, or coordination of nouns or verbs. The second was disagreements caused by (various) phrase structure ambiguities, which a number of them have been discussed in Sect. 2.3.2. The third was the missing or the incorrect labelling of functional tags such as purpose adjunct tag and topical tag.

2.6 Treebank and related tool distribution

The Vietnamese treebank (VTB) and other resources and tools developed by the VLSP project have been posted on the VLSP web page[15] since 2010. They can be used free-of-charge for research purposes. To date, there have been 16,229 visits and 143,920 page views. Statistics for the ten countries that most frequently access the data are shown in Table 10. The current number of online tools used is 140,299, 127,572, and 63,570 for seg-pos-chunk (including word segmentation, POS tagging, and chunking), MRD dictionary, and syntactic parser respectively. Note that the seg-pos-chunk and the syntactic parser were trained by using the VTB. There were about 30 users of VTB (11 from overseas). Of these, 25 were from universities and research institutes and 5 from companies. In order to download the treebank corpus, users were required to agree to use the data for research purposes only (online form).

3 Treebank error detection

3.1 Incorrect tagging, entropy, and classification error

Human languages are complex and ambiguous. In the same context, a linguistic unit (a word, a phrase, a sentence, etc.) should be labelled consistently. The concepts of context and tag depend on NLP problems and data. But the question is how to measure the inconsistency in the data.

[15] http://vlsp.vietlp.org:8080/demo/

Table 10 Top ten country totals

No.	Country	Visits	Percentage
1	Vietnam	12,552	77.34
2	Japan	1389	8.56
3	China	601	3.70
4	Canada	361	2.22
5	United States	264	1.63
6	Singapore	223	1.37
7	France	182	1.12
8	South Korea	160	0.99
9	(not set)	144	0.89
10	Brazil	61	0.38

3.1.1 A motivating example

First, we can consider a motivating example. The following 25-g is a complete sentence that appears 14 times, four times with *centennial* tagged as JJ and ten times with *centennial* marked as NN, with the latter being correct, according to the tagging guidelines Santorini (1990).

– During its *centennial* year, the Wall Street Journal will report events of the past century that stand as milestones of American business history.

Given the PTB data, and given a surrounding context, two words before and twenty two words after, the distribution of *centennial*'s tag over the tag set {JJ, NN} is (4 / 14, 10 / 14). This distribution has a positive entropy value. If all instances of *centennial* were tagged correctly, the distribution of its tags would be (0, 1) and this distribution has an entropy value of zero. This simple analysis suggests that there is a relation between entropy and errors in data, and that high entropy seems to be a problem.

Note that labelled data are often used for training statistical classifiers such as word segmenters, POS taggers, and syntactic parsers. Error-free or reduced-error training data will result in a better classifier. Entropy is a measure of uncertainty. Does an explicit mathematical relation between entropy and classification errors exist?

3.1.2 A probabilistic relationship between entropy and classification error

Suppose that X is a random variable representing information that we know, and Y is another random variable for which we have to guess the value. The relationship between X and Y is $p(y|x)$. From X, we calculate a classification function $g(X) = \hat{Y}$. We can define the probability of error $P_e = P(Y \neq \hat{Y})$. Fano's inequality Cover and Thomas (2006) relates P_e to $H(Y|X)$ as follows:

$$P_e \geq \frac{H(Y|X) - H(P_e)}{log(M-1)} \geq \frac{H(Y|X) - 1}{log(M-1)} \tag{2}$$

where M is the number of possible values of Y. The inequality shows an optimal lower bound on classification-error probability. If $H(Y|X)$ is small, we have more chances to estimate Y with a low probability of error. If $H(Y|X) > 0$, there can be a number of reasons:

- Ambiguity: Y itself is ambiguous, and given X, Y is still ambiguous.
- Choice of X (feature selection): $H(Y) - H(Y|X)$ has been used as information gain in classification studies such as decision tree learning Mitchell (1997).
- Error: For example, the tagging of a word may be inconsistent across comparable occurrences.

In this paper we focus on the relation between $H(Y|X)$ and the correctness of training data. We make two working assumptions:

- There is a strong correlation between high conditional entropy and errors in annotated data.
- Conditional entropy is reduced when errors are corrected.

These assumptions suggest that error correction can be regarded as an entropy reduction process. Now we can consider a more realistic classification configuration using K features rather than only one. Our objective is to reduce the conditional entropy $H(Y|X_1, X_2, ..., X_K)$.

3.1.3 An upper bound of conditional entropy

Since conditioning reduces entropy, it is easy to derive that

$$H(Y|X_1, X_2, \ldots, X_K) \leq \frac{1}{K} \sum_{i=1}^{K} H(Y|X_i) \tag{3}$$

To simplify calculations, instead of directly handling $H(Y|X_1, X_2, \ldots, X_K)$ we can try to reduce the upper bound $\frac{1}{K}\sum_{i=1}^{K} H(Y|X_i)$. Later, through experiments, we will show that this simplification works well. Equation 3 can be straightforwardly proved. Since conditioning reduces entropy Cover and Thomas (2006), we have $H(Y|X) \leq H(Y)$. This inequality implies that on average, the more information there is, the more reduction in uncertainty. By applying this inequality K times, we can obtain $H(Y|X_1, X_2, \ldots, X_K) \leq H(Y|X_i)$ for $1 \leq i \leq K$. Summing up these inequalities and dividing both sides by K, we have Eq. 3.

3.1.4 Empirical entropy

Entropy $H(Y|X_1, X_2, \ldots, X_K)$ can be computed as

$$\sum\nolimits_{x_1, x_2, \ldots, x_K} p(x_1, x_2, \ldots, x_K) \times H(Y|X_1 = x_1, X_2 = x_2, \ldots, X_K = x_K)$$

where the sum is taken over the set $A_1 \times A_2 \times \ldots \times A_K$, A_i are sets of possible values of X_i, and

$$H(Y|X_1 = x_1, X_2 = x_2, \ldots, X_K = x_K) = \sum_y -p(y|x_1, x_2, \ldots, x_K) \times log(p(y|x_1, x_2, \ldots, x_K)).$$

When K is a large number, it is difficult to compute the true value of $H(Y|X_1, X_2, \ldots, X_K)$ since there are $|A_1| \times |A_2| \times \cdots \times |A_K|$ possible combinations of X_i's values. A practical approach to overcome this issue is to compute the empirical entropy of the data set. More specifically, the entropy sum will be taken over $(x_1, x_2, \ldots, x_K)$ for which $((x_1, x_2, \ldots, x_K), y)$ exists in our data set.

In order to compute $p(y|x_1, x_2, \ldots, x_K)$, we need a probabilistic model. A simple approach is to use the Naive Bayes model. Since this model makes a strong independence assumption between X_i, we can decompose $p(y|x_1, x_2, \ldots, x_K)$ into

$$\prod_{i=1}^{K} p(x_i|y) \times p(y) / \prod_{i=1}^{K} p(x_i)$$

where by using the maximum likelihood estimation

$$p(x_i|y) = Freq(y, x_i)/Freq(y), p(y) = Freq(y)/L, \text{ and } p(x_i) = Freq(x_i)/L$$

where L indicates the number of examples in our data set.

Empirical entropy was not used for our error detection methods. It was used only for computing entropy reduction over data sets in Sect. 3.3.6.

3.2 Error detection by ranking

Our method identifies linguistic units (a syllable, a word, a phrase, etc.) whose label varies across the corpus. These units are extracted with tag and context information represented by features whose definition depends on what kind of errors we want to detect. Each occurrence corresponds to an example. We then process this set of variation linguistic units. It can be asked why we do not use sequence models such as n-gram, HMMs or tree structure models such as PCFGs. The reason is that we are currently focusing on the use of entropy for error detection but not complex statistical models. We can observe examples with tag, context, but we do not know which one is erroneous as errors are hidden.

3.2.1 An entropy-based scoring function

Based on the first working assumption stated in Sect. 3.1.2 that "there is a strong correlation between high conditional entropy and errors in annotated data", we rank examples $(x, y) = ((x_1, x_2, \ldots, x_K), y)$ in decreasing order using the following scoring function

$$Score(x, y) = \sum_{i=1}^{K} H(Y|X_i = x_i) + \Delta H \quad (4)$$

where the first term does not depend on y, and the second term ΔH is the maximal reduction of the first term when y varies.

Supposing that B is a set of possible values of Y, $M = |B|$. Without the loss of generality, we may suppose that $B = \{1, 2, \ldots, M\}$. Given that $X_i = x_i$, the discrete conditional distribution of Y is

$$P(Y|X_i = x_i) = (p_1, p_2, \ldots, p_M),$$

where $p_j \geq 0 (1 \leq j \leq M)$ and $\sum_{j=1}^{M} p_j = 1$. Also, p_j can be computed by

$$p_j = Freq(j, x_i)/Freq(x_i)$$

where $Freq(j, x_i)$ is the co-occurrence frequency of j and x_i, and $Freq(x_i)$ is the frequency of x_i which can be easily calculated from a corpus. The conditional entropy can be computed as:

$$H(Y|X_i = x_i) = -\sum_{j=1}^{M} p_j \times log(p_j).$$

When the label of $x = (x_1, x_2, \ldots, x_K)$ changes from y to y', for each x_i, $P(Y|X_i = x_i)$ changes to $P(Y'|X_i = x_i) = (p'_1, p'_2, \ldots, p'_M)$ in which $p'_j = p_j$ for $j \neq y$ and $j \neq y'$, $p'_y = (Freq(y, x_i) - 1)/Freq(x_i)$, and $p'_{y'} = (Freq(y', x_i) + 1)/Freq(x_i)$. The entropy $H(Y|X_i = x_i)$ becomes $H(Y'|X_i = x_i)$ and it is simple to compute ΔH by the formula

$$\begin{aligned}\Delta H &= max_{y'} \sum_{i=1}^{K} [H(Y|X_i = x_i) - H(Y'|X_i = x_i)] \\ &= max_{y'} \sum_{i=1}^{K} \left[-p_y \times log(p_y) - p_{y'} \times log(p_{y'}) + p'_y \times log(p'_y) + p'_{y'} \times log(p'_{y'})\right].\end{aligned}$$

3.2.2 An example of score calculation

Now we take an example to show how the numbers—including probabilities, entropy values, and scores—can be calculated. Our example involves two Vietnamese POS-tagged words, each with ten instances. The first word is 'báo' (as a verb, 'báo' means *to report* or *to inform*; as a noun, 'báo' means *newspaper*). The word has 5 noun instances and 5 verb instances. All instances are correctly tagged. The other word is 'bút' (as a noun, 'bút' means *pen*). This word has 10 noun instances, however, one instance is incorrectly tagged as verb. Table 11 represents a small corpus containing these two words. For simplicity, context is considered to be the previous word only ($K = 1$). We will show that the scoring function $Score(x, y) = H(Y|X = x) + \Delta H$ results in the highest value for the incorrect instance (number 20) of the second word.

In the following calculations, first, for simplicity, we compute $Score'(x, y) = H(Y|X = x)$ (temporarily omit ΔH), where X represents contextual information

Table 11 A small corpus containing POS-tagged instances of 'báo' and 'bút' (in bold). For each instance, the context word is in italic. The sentence number 20 is incorrectly tagged

Number	Sentence
1	Trong *số* **báo**/N thứ 28, chúng tôi có đăng bài của ông.
2	Rồi chàng lôi ở túi áo ra hai *số* **báo**/N Lưỡng Kỳ.
3	Khi xem xong cả hai *số* **báo**/N rồi, lão ta tái mặt, trên trán thấy toát mồ hôi.
4	Einstein, mới 26 tuổi, đã gửi đăng một *bài* **báo**/N trên tạp chí khoa học của Đức.
5	Tổng biên tập của Tạp chí đã thấy rằng *bài* **báo**/N có chất lượng vượt quá mọi yêu cầu.
6	Họ *đã* **báo**/V công an rồi.
7	Trung Quốc *đã* **báo**/V trước ADIZ cho Nhật Bản từ năm 2010.
8	Gí râu móc vào nhau, ấy thế là hai bên *đã* **báo**/V cho nhau đủ tin hoả tốc, rồi lại chạy biến.
9	Chờ cho kẻng hết giờ, *tôi* **báo**/V tin và bảo nó lên xe tôi đi thăm mẹ nó.
10	Bây giờ *tôi* **báo**/V tin cho chị, thuật lại sự việc.
11	Các cụ nhà nho ngày xưa đã *cầm* **bút**/N là phải viết những lời thanh nhã.
12	Vừa *cầm* **bút**/N vài năm nhưng Roth đã có ý thức rất rõ ràng về nghề nghiệp.
13	Có hai viên văn, võ đứng hầu, văn *cầm* **bút**/N đứng tả, võ cầm kiếm đứng hữu.
14	Tự thân không thôi thúc tôi *cầm* **bút**/N viết mà vì sức ép từ thầy của tôi.
15	Người ngoài phải hầu cờ, tiên *cầm* **bút**/N chỉ vào con cờ mà đi từng nước.
16	Nguyễn Trí kể về lần đầu *cầm* **bút**/N của mình.
17	Không ai muốn *đặt* **bút**/N ký thỏa ước ngừng bắn trước cả.
18	Tổng thống của họ đã *đặt* **bút**/N ký trước đức vua nước Hậu Hành.
19	Vì sao Rooney chưa *đặt* **bút**/N ký với M.U?
20*	Như thể không viết được, nhưng lúc đã *đặt* **bút**/V xuống giấy thì nét chữ tươi tắn.

and Y represents tag of the word being considered. Table 12 shows frequency table of (x, y) pairs.

For instances from 1 to 10, since five times 'báo' was tagged with N and five times was tagged with V, the entropy $H(Y)$ reaches the maximal value 1. While for instances from 11 to 20, nine times 'bút' was correctly tagged with N and once was incorrectly tagged with V. The entropy $H(Y) = 0.469$, smaller than 1. However, it does not mean that the first word is more likely to be erroneous than the second word, because we use *conditional entropy* to evaluate error possibility, but not entropy.

For instances from 1 to 16, it is obvious that their context words disambiguate the part of speech well. Considering the first instance (x,y) = (số, N), the conditional probabilities $\mathrm{P(Y{=}N|X{=}số)} = \mathrm{Freq(N,số)/Freq(số)} = 3/3 = 1$, $\mathrm{P(Y{=}V|X{=}số)} = \mathrm{Freq(V,số)/Freq(số)} = 0/3 = 0$, so $p(y|x) = (1, 0)$, $H(p) = 0$, and therefore $Score'(x, y) = 0$. For other instances 2 to 16, similarly we have $Score'(x, y) = 0$.

For instances from 17 to 20, given the context word 'đặt', there are two possible parts of speech, so the conditional distribution $p(y|x) = (3/4, 1/4)$, $H(p) = 0.811$ and therefore $Score'(x, y) = 0.811 > 0$. We can see that this group of instances, including an incorrect one, has a higher score than the others.

Up to now, we can see that for each (x, y) pair, the information about the tag y has not been used. Additionally, four instances from 17–20 have the same score (due to the same x). Now we take ΔH into account: For instances from 17–19, if we change label from N to V: $p'(y|x) = (1/2, 1/2)$, $H(p') = 1$, therefore $\Delta H = H(p) - 1 = -0.189$ For the instance 20, if we change label from V to N: $p'(y|x) = (1, 0)$, $H(p') = 0$, therefore $\Delta H = H(p) - H(p') = H(p) = 0.811$. So, when we take ΔH

Table 12 Frequency table of (x,y) pairs extracted from corpus in Table 11

POS-tagged word	Sentence number	Context x	POS tag y	Frequency of (x,y) pair
báo	1–3	số	N	3
	4–5	bài	N	2
	6–8	đã	V	3
	9–10	tôi	V	2
bút	11-16	cầm	N	6
	17–19	đặt	N	3
	20*	đặt	V	1

into account, the score of the last instance (the incorrect one) increases, while the score of other instances decreases. The incorrect instance has the maximum value of score.

3.2.3 Application to word-segmented and POS-tagged data sets

In this paper, we focus on checking word-segmented and POS-tagged corpora. For word segmented data, syllable n-grams which have multiple word segmentations will be considered (as random variable Y). The features are the two preceding words and the two following words (total of four features, as random variables X_i). For POS tagged data, words with multiple tags are considered. The feature set includes surrounding words and their POS tags (total of eight features). Table 13 shows two examples including labelled sentences, variation n-grams in italics, subscript for mapping Vietnamese-English words, and features.

Table 13 Features for word-segmentation and POS tagging error detection tasks. S1: Word-segmented sentence. S2: POS-tagged sentence. E: English translation

S1: Nguyện_vọng$_1$ về$_2$ vấn_đề$_3$ *nước dùng* đã$_4$ được$_5$ xem_xét$_6$.
E: Proposal$_1$ for$_2$ *clean water* supply$_3$ has$_4$ been$_5$ considered$_6$.
Features: về$_2$, vấn_đề$_3$, đã$_4$, được$_5$
S2: Ông$_1$/N chỉ$_2$/R muốn$_3$/V chui$_4$/V *xuống/E* đất$_5$/N khi$_6$/N chủ_nợ$_7$/N đến$_8$/V./.
E: He$_1$ just$_2$ wanted$_3$ to *disappear*$_{45}$ when$_6$ creditors$_7$ came$_8$./.
Features: muốn$_3$, chui$_4$, đất$_5$, khi$_6$, V$_3$, V$_4$, N$_5$, N$_6$

3.3 Experiments

3.3.1 Corpus description

We can not directly use treebank data for the evaluation of the error-checking task. Dickinson and Meurers (2003) manually checked all instances of variation n-grams to find erroneous instances. However, we did not use Dickinson and Meurers's method. We compared different versions of data sets to find which sentences were modified and at which positions (words or phrases). Table 14 shows the description of the data sets which were used in our experiments. For each data set, two versions were used to extract evaluation data: one version resulting from manual revision, and the other resulting from the second manual revision.

3.3.2 Data extraction

Comparisons were carried out sentence by sentence using minimum edit distance (MED), a dynamic programming algorithm Jurafsky and Martin (2009), in which three operations including insertion, deletion, and replacement are used. The MED algorithm was followed by a post-processing procedure to combine operations for adjacent words of the original sentence. Table 15 shows an example of a word-segmented sentence comparison using the MED algorithm. The underscore character is used to connect syllables of the same word. The syllable sequence *trả giá* is a variation bigram. The MED algorithm found that *trả* (pay) was deleted and *giá* (price) was replaced by *trả_giá* (pay). Since *trả* and *giá* were two adjacent words in the original sentence, deletion and replacement operations were combined together, resulting in the replacement (modification) of *trả giá* by *trả_giá*.

The extraction results of the treebank's two data sets are reported in Table 16. Variation n-grams can be a sequence of syllables with multiple word segmentations in a corpus, or a word with multiple tags in a corpus. An instance (or example) is an occurrence of an n-gram. An erroneous variation n-gram has at least one erroneous instance (incorrectly labelled). This table shows the ambiguous core of the corpus. The percentage of erroneous variation n-grams is high. However the percentage of erroneous instances is much lower. Finding out how to reduce the number of instances to be checked is meaningful.

3.3.3 Error types and distributions

As shown in Table 16, not all instances of variation n-grams are erroneous. Figure 4 displays error distribution curves which show the likelihood of the number of

Table 14 Vietnamese treebank's data sets which were used in experiments

Data set	Sentences	Words	Vocabulary size
1. Word segmented	68,850	1,553,235	45,403
2. POS tagged	10,120	217,111	17,105

Table 15 Example of word-segmented sentence comparison using MED algorithm. S1: erroneous sentence. S2: corrected sentence. E: English translation

S1	Thủ_môn$_1$ *trả giá* vì$_2$ sai_lầm$_3$ ngớ_ngẩn$_4$.
S2	Thủ_môn$_1$ *trả_giá* vì$_2$ sai_lầm$_3$ ngớ_ngẩn$_4$.
E	The goalkeeper$_1$ *pays* for$_2$ his blunder$_{34}$.

Table 16 Data extraction statistics

Data set	Variation n-grams	Error variation n-grams	Instances	Error instances
1	1565	1248	48,752	5227
2	1685	968	108,455	8734

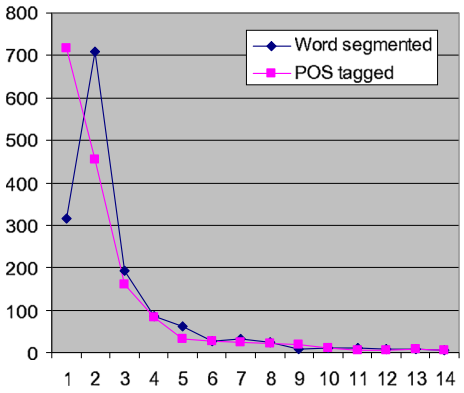


Fig. 4 Error distribution curves. The *horizontal axis* represents error count. The *vertical axis* represents variation n-gram count. The *red curve* corresponds to the word segmentation data set. The *blue curve* corresponds to the POS tagged data set

erroneous instances of a variation n-gram. These curves look like Poisson distributions. The Poisson distribution is typical for rare events. For the word-segmented data set, on average each variation n-gram has 31.15 instances in total and 3.34 erroneous instances. For the POS-tagged data set, on average each variation n-gram has 64.36 instances in total and 5.18 erroneous instances. Maximum points are close to the vertical axis.[16] It is clear that most variation n-grams have zero, one, two, or several errors.

In the word segmented data set, about 60 % of erroneous instances require correction by combining single words to form a compound word. About 40 % require a change by splitting a compound word into single words. A number of typical corrections are listed here: subordinated compound

[16] Two points nearest to the vertical axis are the number of variation n-grams which have no erroneous instances.

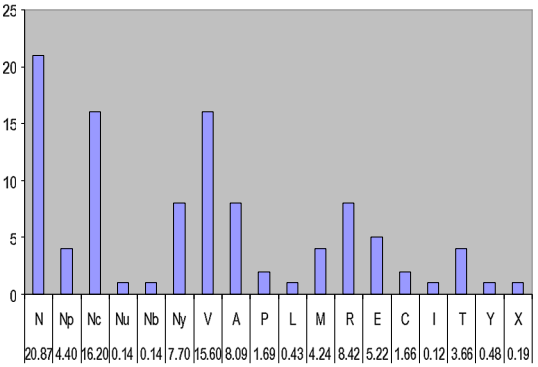


Fig. 5 The percentage of each corrected POS tag

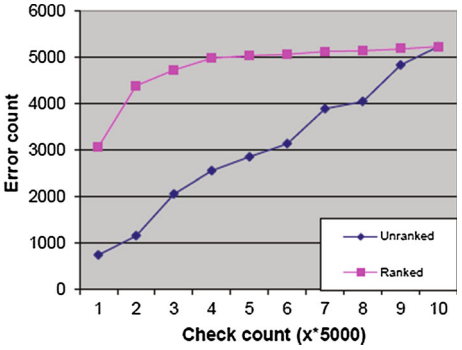


Fig. 6 Error detection results for word segmentation. The *horizontal axis* represents the number of examples annotators have to check. The *vertical axis* represents the number of erroneous examples

((khu phố → khu_phố (quarter), kim khâu → kim_khâu (needle)), coordinated compound (thu đông → thu_đông (autumn and winter), xinh đẹp → xinh_đẹp (beautiful)), another kind of subordinated compound (nhà khoa_học → nhà_khoa_học (scientist), nguyên bộ_trưởng → nguyên_bộ_trưởng (former minister)), proper noun (Công_ty_FPT → Công_ty FPT (FPT company), Hà Nội → Hà_Nội).

Figure 5 shows the percentage of each modified POS tag. For example, the first column shows that among 8734 (Table 4) erroneous POS tagged instances, 20.87 % were changed from the noun tag N to other POS tags. Of 18 columns, the ones corresponding to noun, verb, adverb, and adjective have the largest percentage.

3.3.4 Error detection results for word segmentation

Figure 6 shows error detection results for word segmentation. The blue curve represents the number of erroneous examples discovered when annotators check the data set in which examples are in the original order. The red curve represents the number of erroneous examples discovered if annotators check the data set in which examples are sorted in decreasing order of entropy. It is obvious that most errors, about 89.92 % (4700/5227) have been detected after checking one third of the data set.

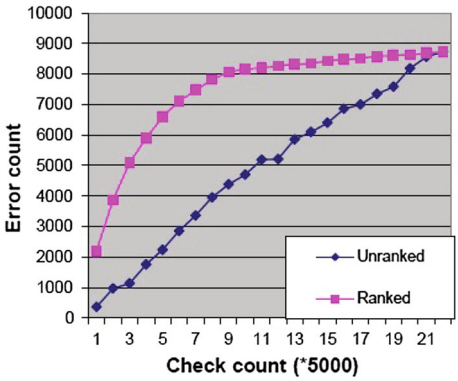


Fig. 7 Error detection results for POS tagging. The *horizontal axis* represents the number of examples annotators have to check. The *vertical axis* represents the number of erroneous examples

3.3.5 Error detection results for POS tagging

Figure 7 reports error detection results for POS tagging. If annotators check data with examples in original order, the number of detected errors goes up linearly (blue curve). If the data is sorted in decreasing order of entropy, the number of detected errors goes up very fast (red curve), about 81.34 % (7104/8734) after checking of one third of the data set.

3.3.6 Entropy reduction

Entropy plays a central role in our detection method. High entropy corresponds to high possibility of errors. Table 17 shows that on both data sets, total empirical entropy of all variation n-grams has already been reduced after error correction (EntDecTotal). Also, total entropy upper bound has also decreased (EntBDecTotal). For the word-segmented data set, a majority of erroneous n-grams (92.90 %) shows less entropy after error correction, a very small number show no change (0.97 %) in entropy, and 6.13 % show increasing entropy. For POS-tagged data set, the percentage of increased-entropy erroneous n-grams is higher.

According to our observations on specific erroneous n-grams, there are a number of reasons for the increase of entropy. The first is the sparse data problem. For n-grams with a small number of instances and few errors, the correction of errors leads to an entropy increase in some cases. The second is that some words are highly ambiguous, and after revision there are still errors. Within the set of 95 erroneous n-grams whose number of erroneous instances is greater than 15, there are 39 n-grams (41.05 %) whose entropy increased. Though this is a small set, the ratio is high in comparison with 22.71 % on average.

It is logical that the entropy upper bound is reduced more than empirical entropy. However, it seems that the difference between these values is rather large. Note that empirical entropy is summed over a subset of the whole space, so it is smaller than

Table 17 Entropy changes on data sets (DS). EntDec/EntUnc/EntInc n-gram: the percentage of erroneous n-grams for which entropy decreased/remained/increased; EntDec Total: total entropy reduction of n-grams; EntBDec Total: total entropy bound reduction of n-grams

DS	EntDec n-gram (%)	EntUnc n-gram (%)	EntInc n-gram (%)	EntDec Total	EntB DecTotal
1	92.90	0.97	6.13	11.62	82.90
2	76.96	0.33	22.71	13.22	69.17

the true entropy value. If $p(x_1, x_2, \ldots, x_K)$ is normalized, the calculation of empirical entropy reduction will result in a higher value.[17]

In the image processing research field, there was a related work on image restoration Awate and Whitaker (2006). The purpose of image restoration is to "undo" defects which degrade an image. That paper proposed an unsupervised, information theoretic method that improves the predictability of pixel intensities from their neighborhoods by decreasing their joint entropy. This method can automatically discover the statistical properties of the signal and can thereby restore a wide spectrum of images. The paper describes a gradient-based technique of minimizing the joint entropy measure and presents several important practical considerations in estimating neighborhood statistics. Experiments on both real and synthetic data, along with comparisons with current state-of-the-art techniques, showed the effectiveness of this entropy-based method.

4 Conclusions

In this paper, we have reported on the construction of the first large-scale Vietnamese treebank. Since this work is interdisciplinary between natural language processing and linguistics, we briefly focused on linguistic solutions and controversial issues concerning our annotation schemes. Such information may be useful for other languages with the typology similar to Vietnamese, and also useful for researchers and users of this treebank. Though our national project is officially finished, we will continue revising data through syntactic phenomena and feedback from users. We intend to publish these data with the LDC in the near future.

We have investigated an entropy-based method for detecting errors and inconsistencies in the word-segmented and POS-tagged parts of our treebank data. Our experiments have shown that this method is effective. More specifically, it can reduce the size of error candidate sets by two thirds, and significantly reduce conditional entropy after correction of errors. In the future, we intend to apply the entropy-based approach to detecting syntax tree errors, and to use additional resources, such as word clusters, to improve our error detection results.

[17] Using $p(x_1, x_2, \ldots, x_K) = Freq(x_1, x_2, \ldots, x_K)/L$, the value of empirical entropy reduction was 173.49 on the word-segmented data set.

Acknowledgments This paper is supported by the project QGTĐ.12.21 funded by Vietnam National University, Hanoi. We would like to express special thanks to other members of the treebank development team Xuan-Luong Vu and Dr. Thi-Minh-Huyen Nguyen, and linguistic annotators Minh-Thu Dao, Thi-Minh-Ngoc Nguyen, Kim-Ngan Le, Mai-Van Nguyen for the effective cooperation. We also would like to express thanks to Assoc. Prof. Dinh Dien for his comments and discussions during the early stages of the treebank development.

References

Awate, S. P., & Whitaker, R. T. (2006). Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*, 364–376.

Berger, A., Pietra, S. D., & Pietra, V. D. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics*, *22*(1), 39–71.

Black, E., Abney, S., Flickenger, D., Gdaniec, C., Grishman, R., Harrison, P., et al. (1991). A procedure for quantitatively comparing the syntactic coverage of English grammars. In *Proceedings of DARPA speech and natural language workshop*.

Cao, X.-H. (2007). *The Vietnamese language: Phonetics, syntax, and semantics [in Vietnamese]*. Cambridge: Education Press.

Chiang, D., & Bikel, D. M. (2002). Recovering latent information in treebanks. In *Proceedings of COLING*.

Collins, M. (1999). *Head-driven statistical models for natural language parsing*. Ph.D. thesis, University of Pennsylvania.

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory*. New York: Wiley.

Dickinson, M., & Meurers, W. D. (2003). Detecting errors in part-of-speech annotation. In *Proceedings of EACL*.

Dickinson, M. (2006). From detecting errors to automatically correcting them. In *Proceedings of EACL*.

Dickinson, M. (2008). Ad hoc treebank structures. In *Proceedings of ACL*.

Diep, Q.-B. (2005). *Vietnamese syntax [in Vietnamese]*. Cambridge: Education Press.

Han, C., Han, N., Ko, E., & Palmer, M. (2002). Development and evaluation of a Korean treebank and its application to NLP. In *Proceedings of LREC*.

Johnson, M. (1998). PCFG models of linguistic tree representation. *Computational Linguistics*, *24*, 613–632.

Jurafsky, D., & Martin, J. H. (2009). *Speech and language processing: An introduction to natural language processing*., Computational linguistics and speech recognition New Jersey: Prentice Hall.

Klein, D., & Manning, C. D. (2003). Accurate unlexicalized parsing. In *Proceedings of ACL*.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*.

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, *19*, 313–330.

Mitchell, T. M. (1997). *Machine learning*. Maidenhead: McGraw-Hill.

Miyao, Y., & Tsujii, J. (2008). Feature forest models for probabilistic HPSG parsing. *Computational Linguistics*, *34*, 35–80.

Nguyen, V.-H. (2009). *Vietnamese syntax [in Vietnamese]*. Cambridge: Education Press.

Nguyen, T.-M.-H., Vu, X.-L., Le, & H.-P. (2003). A case study of the probabilistic tagger QTAG for tagging Vietnamese texts [in Vietnamese]. In *Proceedings of ICT.rda*.

Nguyen, T.-C. (2004). *Vietnamese syntax [in Vietnamese]*. Hanoi: Vietnam National University Press.

Nguyen, P.-T., Vu, X. L., Nguyen, T. M. H., Nguyen, V. H., & Le, H. P. (2009). Building a large syntactically-annotated corpus of Vietnamese. In *Proceedings of LAW-3, ACL-IJCNLP*.

Nguyen, V.-H. (2009). The history of approaches in describing Vietnamese syntax. *Journal of the Research Institute for World Languages*, (1), 19–34

Novak, V., & Razimova, M. (2009). Unsupervised detection of annotation inconsistencies using apriori algorithm. In *Proceedings of LAW-3, ACL-IJCNLP*.

Pajas, P., & Stepanek, J. (2008). Recent advances in a feature-rich framework for treebank annotation. In *Proceedings of COLING*.

Phuong, L. H., Huyen, N. T. M., Azim, R., & Vinh, H. T. (2008). A hybrid approach to word segmentation of vietnamese texts. In *Proceedings of the 2nd international conference on language and automata theory and applications*. Springer LNCS 5196, Tarragona, Spain, 2008.

Rambow, O. (2010). The simple truth about dependency and phrase structure representations: An opinion piece. In *Proceedings of NAACL*.

Santorini, B. (1990). Part-of-speech tagging guidelines for the Penn Treebank Project. In *Treebank-3 Documents*. Linguistic Data Consortium.

Sciullo, A. M. D., & Williams, E. (1987). *On the definition of word*. Cambridge: The MIT Press.

Steedman, M., Osborne, M., Sarkar, A., Clark, S., Hwa, R., Hockenmaier, J., et al. (2003). Bootstrapping statistical parsers from small datasets. In *Proceedings of EACL*.

Thompson, L. C. (1987). *A Vietnamese reference grammar*. Hawaii: University of Hawaii Press.

van Halteren, H. (2000). The detection of inconsistency in manually tagged text. In *Proceedings of LINC*.

Xue, N., Xia, F., Chiou, F.-D., & Palmer, M. (2005). The Penn Chinese TreeBank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, *11*, 207–238.

Yamada, H., & Matsumoto, Y. (2003). Statistical dependency analysis with support vector machines. In *Proceedings of IWPT*.

Yates, A., Schoenmackers, S., & Etzioni, O. (2006). Detecting parser errors using web-based semantic filters. In *Proceedings of EMNLP*.