Evaluating Parts-of-Speech Taggers for Use in a Text-to-Scene Conversion System

Kevin Glass and Shaun Bangay
Computer Science Department
Rhodes University
Grahamstown, South Africa

This paper presents parts-of-speech tagging as a first step towards an autonomous text-to-scene conversion system. It categorizes some freely available taggers, according to the techniques used by each in order to automatically identify word-classes. In addition, the performance of each identified tagger is verified experimentally. The SUSANNE corpus is used for testing and reveals the complexity of working with different tagsets, resulting in substantially lower accuracies in our tests than in those reported by the developers of each tagger. The taggers are then grouped to form a voting system to attempt to raise accuracies, but in no cases do the combined results improve upon the individual accuracies. Additionally a new metric, *agreement*, is tentatively proposed as an indication of confidence in the output of a group of taggers where such output cannot be validated.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Language parsing and understanding*

General Terms: Languages, Measurement, Verification

Additional Key Words and Phrases: Corpora, Parts-of-speech tagging

1. INTRODUCTION

1.1 Problem Statement

Text-to-scene conversion requires the interpretation of written text in order to present a graphical representation of the content. The first step in such interpretation is the accurate identification of the word-class of each word, otherwise known as parts-of-speech tagging . A number of techniques exist that automatically determine parts-of-speech in free text, including probabilistic, tree-based, rule-based, maximum-entropy, and support vector machine-based techniques. However, majority of these techniques are evaluated using a single set of test data (the Wall Street Journal section [Brill 1994; Toutanova et al. 2003; Giménez and Marquez 2003] of the Penn Treebank corpus [Marcus et al. 1994]), and thus accuracy results are unsuitable for the domain of text-to-scene conversion which addresses a large variety of writing styles (specifically found in fictional writing). In addition, no evidence can be found of a combination of these techniques in voting-systems aimed at improving accuracy. Since large amounts of unannotated text will be used in text-to-scene conversion, determining the accuracy of the assigned word-classes is also a problem, and a metric which can measure confidence in the output of a given ensemble would be useful in evaluating the extent to which incorrect word-classes affect the incorrect rendering of a scene. Therefore, the problems addressed in this paper are as follows:

—Validate the accuracy of various parts-of-speech tagging techniques on a less frequently used corpus;
—Combine the parts-of-speech taggers in various ways to improve accuracy;
—Investigate a metric, *agreement,* as an indication of confidence in the output of an ensemble of parts-of-speech taggers.

1.2 Background

The increased capacity and power of modern computers has resulted in an opportunity for storing and presenting text in novel ways. Digitization of books presents many opportunities for computational study of natural language, particularly in fictional literature. One such opportunity is in the field of virtual reality. WordsEye [Coyne

Kevin Glass, Computer Science Department, Rhodes University, P O Box 94, Grahamstown, 6140, South Africa; kglass@rucus.net
Shaun Bangay, Computer Science Department, Rhodes University, P O Box 94, Grahamstown, 6140, South Africa; s.bangay@ru.ac.za

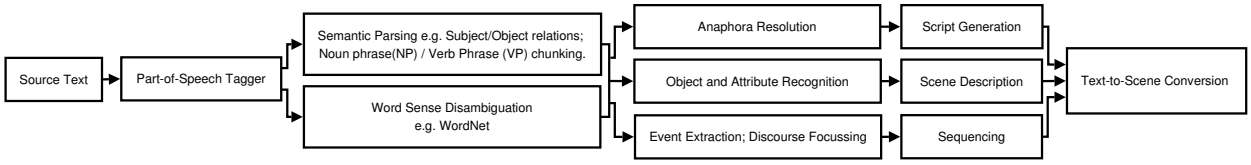

Figure 1. The Parts-of-speech tagger as a foundation to a text-to-scene conversion system.

and Sproat 2001], for example, is a system that converts simple natural language descriptions into 3-dimensional images. The CONFUCIUS system [Ma 2002] aims at automatically converting books into a multi-modal presentation (that is, including visual and audio modes).

Our research aims to determine to what extent the book can be used to provide information regarding the layout of a three dimensional scene. It is our hypothesis that electronic books may provide an opportunity to create rich virtual environments with greater ease than current manual systems allow.

Deriving meaning from natural language is an ambitious goal. However, there are currently several aspects of text-to-scene conversion which we believe can benefit from automation. Such aspects include:

—**Verbal Discourse (script generation)**: correctly extracting direct-speech from the text, and associating the speech with the correct speaker. The desired result would be a script which can be easily understood by a speech synthesis system. A key problem here is the identification of the correct speaker when indirect referencing is used, such as the pronoun *he* in *he said*. This problem falls under the wider field of *pronominal anaphora resolution*.

—**Scene Description:** the extraction of useful information regarding the contents, layout and attributes of objects/characters in the scene.

—**Sequencing:** constructing a coherent order of events that can be reconstructed in a virtual environment. A key problem here is identifying changes in context which may signal the creation of a new scene.

As a first step the simplest semantic meaning from written text needs to be identified, that is, the word-class, or part-of-speech of the words. The grouping of words into categories including nouns, verbs and adjectives, provides enough information to create a simple list of renderable objects in a scene, including their attributes and relative positions. Once the labelling is achieved the words can be fed into a parsing system which provides deeper relationships between groups of words, such as the *subject/object* of a sentence. Such relations can be used further, for example, in disambiguating the referent of a pronoun. Figure 1 illustrates the importance of a parts-of-speech tagger (henceforth referred to as POS tagger) in a text-to scene system.

1.3 Overview of the Paper

The rest of this article follows the following format: A brief illustration of the applicability of a POS tagger in anaphora resolution is given in section 2.1. Section 2.2 briefly discusses the problem of POS tagging, and Section 2.3 provides brief explanations of the workings of a number of available POS taggers, each which implement a different technique. Finally section 3 presents some initial experimental work with the available POS taggers.

2. RELATED WORK

2.1 Parts-of-Speech Tagging in Anaphora Resolution

Anaphora resolution refers to the process of identifying when words or phrases in text refer to the same entity. For example, in the sentence, *'Cathy was very tired, and her eyelids began to droop,'* the word *her* refers to the entity named Cathy.

Much work has been done to develop algorithms which will accurately find the correct referent *(antecedent)* of a pronoun. In particular, a number of knowledge-poor (that is, without semantically parsing a sentence) approaches exist [Mitkov et al. 2002; Dimitrov 2002; Kennedy and Boguraev 1996], which rely on the correct identification of word-classes such as pronouns. Semantic parsing can also be used for anaphora resolution [Hobbs 1978], and is dependent on accurate parts-of-speech tagging.

2.2 Parts-of-Speech Tagging

The word-class, or tag, of the word is highly dependent on the context in which it is used. For instance, the word *man* can be placed into two word-classes: noun, in the sentence, 'The man was walking'; and verb in, 'He will man the lifeboat'. Consequently, most autonomous POS taggers employ some kind of learning system which is able to formulate knowledge about the likelihood of a certain tag occurring, given its context.

Name	QTag	TreeTagger	Brill	Stanford	SVMTagger
Type	Probabilistic	Decision Tree	Rule-based	Maximum Entropy	Support Vector Machine
Reported Accuracy (%)	98.39	96.36	97.20	97.24	97.20

Table 1. Parts-of-speech taggers discussed in this article

In order to establish this contextual knowledge, a process of training must occur. One requirement for this training is a large enough set of words which has already been classified. This type of information can be found in a resource known as an *annotated corpus.* An example of such a corpus is the Penn Treebank [Marcus et al. 1994], which consists of one million words, manually annotated, from the Wall Street Journal. It is the most popular corpus for parts-of-speech tagging, and is widely used as a source of benchmarking data for evaluating new POS tagging techniques.

Typically, 80% of the corpus is used to train a tagger, and the remaining 20% used as test data. This test data is then compared to the original annotations, provided with the corpus, in order to determine the tagger's accuracy.

Of major importance in a POS tagger is its *tagset,* which is a list of representative symbols for each word-class. Differing corpora make use of differing tagsets, but because of the popularity of the Penn Treebank, its corresponding tagset is the most widely used, and contains a total of 48 types of tags.

This article discusses and evaluates five different classes of freely available POS taggers. These taggers are listed in Table 1, and are discussed in more detail in Section 2.3.

2.3 Overview of Available Parts-of-Speech Taggers

2.3.1 *Probabilistic Tagger*

Conventional POS taggers, otherwise known as *ngram-taggers*, are probabilistic taggers which look at $n-1$ words previous to the current word in order to establish its word-class. The QTag software [Tufis and Mason 1998] implements a window of three words (*trigram*), where the probability of each possible tag for a current word is combined with the likelihood that the tag is preceded by the two previously assigned tags. The tag with the highest score is selected. The initial probabilities are calculated from a training corpus.

Because of the statistical nature of probabilistic taggers, they may be trained for various languages. QTag has only undergone one definitive test, in which it was tested on a Romanian corpus of approximately 250 000 words. Accuracies of between 95.63% and 98.39% are reported. The version used in the testing (Section 3) is trained for English.

2.3.2 *Decision Tree-based Tagger*

The TreeTagger [Schmid 1994] makes use of a binary decision tree in order to estimate probabilities. The difference between conventional *ngram* taggers and the TreeTagger is that while conventional *ngram* taggers estimate the probability using a maximum likelihood principle, the TreeTagger estimates probabilities with a binary decision tree. In order to determine the probability of a given trigram, a path is followed through a decision tree until a leaf node containing such probabilities is reached.

The TreeTagger was tested on 100 000 words from the Penn Treebank (sourced from differing areas to those used for training), and compared to conventional trigram taggers. An accuracy of 96.34% (trigram context) and 96.36% (quatrogram context) is reported.

2.3.3 *Rule-based Tagger*

The Brill tagger uses a rule-based approach [Brill 1994], where a set of rules for determining word tags is created as follows (during training): an initial set of naive tags are assigned to the corpus of words, after which *transition rules* are learned by correcting the falsely identified wordtags. During the tagging process, these rules are applied in order to identify the correct word tag.

The Brill tagger was trained on 600 000 words from the Penn Treebank corpus, and tested on a separate 150 000 words from the same corpus, achieving an accuracy of 97.2%.

2.3.4 *Maximum Entropy Tagger*

Stanford University have implemented a tagger using a *maximum entropy approach* [Toutanova and Manning 2000]. In information theory, entropy is defined as the average quantity of information generated by some information generating function [Schwartz 1963]. This function has the property that the less the probability of a certain state occurring, the more information is generated when such a state occurs. For a given word and its context, every tag in the tagset is assigned a probability based on data derived during training. Hence the

	WSJ	SUSANNE
Number of Words	Approx. 1 170 000 [Giménez and Marquez 2003]	130 000 [Sampson 2005]
Content	Press reportage	Press reportage, letters, biographies, memoirs, technical writing, adventure and Western fiction [Sampson 2005]
Size of Tagset	48 [Marcus et al. 1994]	353 [Sampson 2005]

Table 2. Major differences between the Wall Street Journal (WSJ) section of the Penn Treebank corpus and the SUSANNE corpus.

probability of a tag sequence can be calculated for a sequence of words, resulting in a probability distribution. A number of distributions can be generated by analysing different *features* in the text. The distribution with the highest entropy, or information gain, is chosen.

Additionally, a *bidirectional* approach is used for determining the context of a word (as opposed to the *unigram* approach of *ngram*-taggers), looking both before and after a word for clues regarding its word-class [Toutanova et al. 2003].

This tagger was tested on the Penn Treebank, yielding accuracy rates of up to 97.24% using this training set.

2.3.5 *Support Vector Machine Tagger*

A learning technique known as Support Vector Machines (SVM) has also been used for POS tagging [Giménez and Marquez 2003]. This technique is used for binary classification, with the aim of learning a *linear hyperplane* that separates a set of positive examples from negative examples with a maximal margin. In a word-tagging context, binary classification is achieved by training a SVM for each part-of-speech in order to distinguish between classes. A dictionary is extracted from a training corpus with all possible tags for each word. Each word w tagged as t_i in the training corpus provides a positive example for tag t_i and a negative example for all other tag classes t_j. When deciding which tag to assign to a word, the most confident tag according to the predictions of all the binary SVMs is selected. A centered window of seven tokens is used, which is larger than the common window of three tokens in a *trigram* tagger.

A number of experiments were done with the SVM tagger, based on the Penn Treebank data. The accuracy of this tagger is reported to be 97.2%.

2.3.6 *Combination of Parts-of-Speech Taggers*

Combinations of POS tagging systems have also been implemented with the aim of improving overall accuracy. One of the first studies on the combination of POS taggers is demonstrated by van Halteren *et al.* [1998], where various combination techniques are used, ranging from simple voting and pairwise voting, to the *stacking* of the classifiers, and second level learners. Brill and Wu [1998] postulate that although each tagger uses the same contextual information regarding the current word in order to define its tag, each one makes use of it in a different manner. In both articles a marked improvement in accuracy is achieved. In Márquez *et al.* [1999] several types of classifiers are combined into decision tree-based ensembles. These ensembles are constructed using machine learning techniques. Once again results are comparable to individual available POS taggers.

3. EXPERIMENTATION

These experiments have the following goals:

—**Validation:** to validate the reported accuracies of the discussed POS taggers on a corpus other than the Penn Treebank.

—**Combination:** to determine if a higher accuracy can be achieved by combining this novel collection of POS taggers into a voting system.

—**Agreement:** to determine if the confidence in choice of tag can be measured when using combination of POS taggers, in the absence of an annotated corpus.

3.1 Validation

3.1.1 *Method*

The aim of this experiment is to validate the accuracy of each of the above mentioned POS taggers. The SUSANNE corpus [Sampson 2005] is used for accuracy testing and consists of 130 000 words extracted from the Brown corpus. This corpus is sufficiently different from the Penn Treebank corpus to warrant its use in this experiment (see Table 2). The raw text extracted from the SUSANNE corpus is processed by each tagger

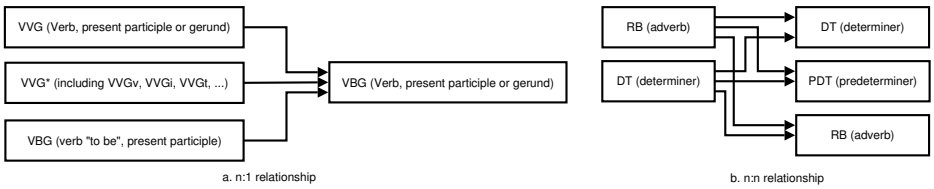

Figure 2. Illustration of a $n:1$ relationship (a), and a $n:n$ relationship (b).

	Reported	57 tags	19 tags	TOs remapped
Q	98.39	62.36	89.28	90.35
T	96.36	60.44	93.47	94.53
B	97.20	72.72	93.18	94.25
E	97.24	71.57	90.60	91.67
S	97.20	72.88	93.24	94.31

Table 3. Comparative results of the performance of various POS taggers on the SUSANNE corpus. Q (QTag); T (TreeTagger); B (Brill); E (Maximum Entropy); S (Support Vector Machine).

(the taggers are not retrained), and the outputs compared to the corresponding tags in the corpus, in order to evaluate accuracy.

3.1.2 *Implementation*

The major difficulty in conducting this experiment is that the SUSANNE corpus makes use of a non-standardised tagset, which needs to be translated to the Penn-style tagset so that the accuracy of the various taggers can be ascertained. This mapping process between the two tagsets is done manually, reducing the 353 large SUSANNE tagset down to a 57 large tagset (punctuation tags are ignored). However, mappings between the larger and the smaller tagsets prove to be $n:n$ in nature, and not $n:1$ as expected (Figure 2). As a result, any sentences containing tags which can not be directly mapped are ignored, reducing the word count to approximately 70 000 words.

The resulting mapping causes the accuracy of the various taggers to drop substantially, because subtle differences in the meanings of the tags cannot always be accurately translated. In response to this, the tagsets are further minimised, to only encompass the base parts of speech (for example, any type of noun - singular, plural, proper - are mapped to a single noun class). This raises the accuracies of the taggers substantially.

3.1.3 *Results*

Table 3 presents the results of the various taggers on the abridged SUSANNE corpus. It is noticeable that the more generalised the tagset becomes, the more accurate the tagging process becomes.

Table 3 also indicates that reducing the size of the tagset from 57 to 19 tags improves the performance of each tagger by at least 20%. During testing it was noted that a large amount of error was caused by the incorrect tagging of the word "to". Consequentially, all instances of this word are marked as TO, which raises the accuracies of the taggers by more than 1% in all cases. Finally, it can also be seen that the accuracies of the taggers on the SUSANNE corpus are between 1.83% and 8% lower than reported in the taggers' corresponding articles.

3.1.4 *Conclusions*

The accuracies observed in this experiment are not as high as those reported for each tagger. A reason could be that most taggers are trained and tested on the Penn Treebank corpus, while in this case the taggers are trained and tested on two different corpora. This may account for some loss in accuracy. Also the translation between the SUSANNE and Penn Treebank tagsets might introduce an element of error. Finally, a large amount of the corpus is removed because of unmappable tag translations. We speculate that words removed for this reason would easily have been correctly tagged, because of their uniqueness. Hence overall accuracy is reduced. According to this test the TreeTagger is the most accurate, followed by the Brill tagger and the SVMTagger.

3.2 Combination

3.2.1 *Method*

The purpose of this experiment is to determine if more accurate tagging can be achieved by combining the different parts-of-speech taggers. Four types of combination are proposed:

	Simple	Weighted	Ranked	Optimal
QTEBS	93.82	93.81	93.92	93.95
TBES	93.81	94.21	94.27	94.27
TBS	94.32	94.34	94.34	94.34
TS	94.11	94.43	94.43	94.43
TB	94.10	94.09	94.09	94.10
QBES	93.31	93.52	93.55	93.84

Table 4. Accuracies achieved by combining taggers.

	Q	T	E	B	S
QTEBS	0.0348	1	0.0372	0.1034	0.9582
TBES	-	1	0.4171	0.4605	0.5892
TBS	-	1	-	0.9316	0.1602
TS	-	1	-	-	0.2035
TB	-	1	-	0.8217	-
QBES	0.6819	-	0.0060	1	0.7748

Table 5. Optimal weightings found for combinations of taggers using a genetic algorithm.

(1) **Simple Vote:** the tag which is selected by majority of the taggers is chosen. In the case of a tie, a random tag between the tied parties is chosen.

(2) **Weighted Vote:** the vote of each tagger is weighted based on the performance of the tagger in Section 3.1. The tag which achieves the highest score is chosen.

(3) **Ranked Vote:** each of the taggers is given a rank (between 1 and 5) based on the performance in Section 3.1, with the best scoring tagger assigned a rank of five. Tag scores are calculated by adding the ranks of the taggers which voted for each specific tag. The tag which achieves the highest score is chosen.

(4) **Optimal Weighted Vote:** an optimum weighting between taggers is calculated using a genetic algorithm.

3.2.2 *Implementation*

Each tagger is given the opportunity to tag the text from the SUSANNE corpus. This tag information is then combined as explained in Section 3.2.1 to determine a tag for each word. The resulting tag is then compared with the original tag from the SUSANNE corpus in order to establish accuracy.

3.2.3 *Results*

Table 4 presents the accuracies of the various taggers in combination with each other. In each row the tagger which performs the worst (based on Section 3.1) is removed. In no cases are the combined results greater than the highest individual accuracy as presented in Table 3. Using a simple voting scheme yields accuracies which appear to be midway between the best and worst individual tagger. As the worst performing taggers are removed the accuracies increase. Weighting and ranking both produce slight improvements in accuracy, and appear to have moderate results on improvement only when there are more than three taggers competing. The search for optimal weightings for each tagger results in slight improvements. Table 5 presents the weightings for a combination of taggers which produce the highest results. Generally, the tagger with the lowest weighting corresponds with the tagger to be dropped, except when dropping B from TBS. Thus the TB combination is also tested, but does not show any improvement in accuracy on the TS combination. Finally, a combination excluding the TreeTagger is tested, which also fails to better the performance of all its component taggers.

3.2.4 *Conclusions*

Although none of the voting systems outperformed the most accurate individual tagger (TreeTagger), the voting system did manage to produce higher accuracies than all of the other taggers available. However, the fact that accuracies increase as taggers are removed indicates that combining taggers reduces accuracy rather than improves it.

Combinations between the three most accurate taggers TreeTagger, Brill tagger and SVMTagger produce noticeably higher accuracies than when the other two taggers are included. Also there is less variation in accuracy between different voting schemes when these taggers are used. This reinforces the conclusion that these are the three most accurate taggers.

3.3 Agreement

Practical text-to-scene systems will use text with different characteristics to the annotated corpora which are available. No suitable corpus exists to allow testing of POS tagging on such text. Thus a metric is required that can evaluate combinations of taggers on unannotated text.

Agreement is proposed as a metric which aims to measure the confidence one can have in the output of a group of taggers. Ideally this metric will work on data which cannot be validated for accuracy. It is hypothesised that there exists a relationship between accuracy and agreement.

3.3.1 *Method*

Three types of agreement are compared in this experiment in order to evaluate how often the taggers concur:

(1) **Simple Agreement**: for each word, the highest vote is divided by the total number of votes. These ratios are then averaged over all the words tagged.

(2) **Total Agreement:** agreement is calculated as the average number of unanimous votes (all taggers vote for the same tag).

(3) **Pairwise Agreement:** for each word the number of pairs of agreeing tag votes are summed and divided by the maximum number of agreeing pairs. The result over all words is summed and averaged. The number of agreeing pairs in a vote is a combinatorial problem, where n is the number of taggers, with pairs of 2 being chosen:

$$\alpha = \binom{n}{2} = \frac{n!}{2!(n-2)!} = \frac{n(n-1)(n-2)!}{2!(n-2)!} = \frac{n(n-1)}{2}$$

The above calculation is done for each tag nominated for a word (α_{TAG}), where n is the number of votes obtained for the tag. The sum of agreeing pairs for a word is calculated, and divided by the total agreement, where n is is the number of taggers available (α_{TOT}). Thus if w words are evaluated, with i tags associated with word x and n taggers are available, then agreement is defined as:

$$agreement = \frac{1}{w}\sum_{x=0}^{w}\frac{\sum_{y=0}^{i}\alpha_{TAG(y)}}{\alpha_{TOT}}$$

Pairwise agreement reflects subtleties in agreement that simple agreement cannot. For instance, if 3 taggers vote for a noun, while 2 vote for a verb, then the pairwise agreement is 0.4, while the simple agreement is 0.3. The pairwise agreement yields more information, since the agreement between the two non-winning votes is also taken into account.

In order to establish a relationship between agreement and accuracy, an experiment is done where copies of the SUSANNE corpus are made with random modifications in order to reduce accuracy. The agreement and accuracy metrics are then calculated by comparing the less accurate corpora against the original corpus.

3.3.2 *Implementation*

Five instances of the SUSANNE corpus are created and corrupted to an extent specified by an accuracy probability. Corruption takes the form of randomly choosing a tag for a word from the Penn tagset. This simulates five taggers that are either correct, or chose a tag randomly. Ideally a tag would be chosen from a subset of legal tags for the specific word, but in this case any tag from the entire tagset may be chosen. Therefore this serves as a worst case scenario for the agreement metric. Initially the specified accuracy probability is small, and is iteratively increased, with simple accuracy and agreement measured at each iteration.

The agreement metrics are also tested on a large amount of fictional free text (approximately 350 000 words) from the Gutenberg corpus (not annotated). No accuracy information is available, but confidence in the results can be ascertained by looking at the agreement.

3.3.3 *Results*

Figure 3 illustrates the relationship between accuracy and agreement (error-bars are included, but have minimal significance). All three agreement metrics show a direct relationship with accuracy.

Table 6 presents the agreement metrics after tagging unrestricted text, and reflects an important anomaly. Table 4 indicates that as taggers are removed from the combination the accuracies steadily increase. If a direct relationship exists between accuracy and agreement (as in Figure 3) then the agreements in Table 6 should also steadily increase. This is the case for total agreement, but for the TBS and TS combinations pairwise and simple agreement fail to show progressive improvement.

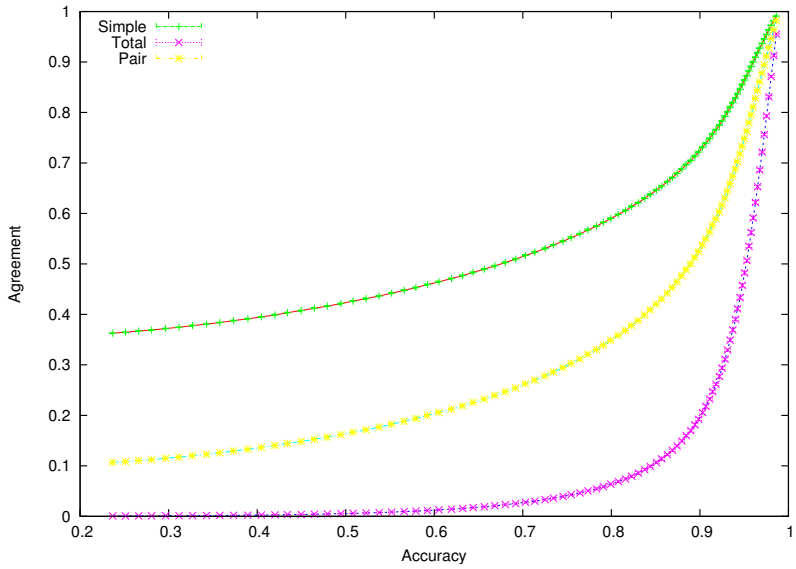

Figure 3. Relationship between accuracy and agreement

	Simple	Total	Pair
QTBES	91.65	73.22	85.56
TBES	92.79	80.61	88.03
TBS	93.36	83.96	88.01
TS	92.92	85.84	85.84

Table 6. Agreement metrics calculated on unannotated data (with notation as in Table 3).

3.3.4 *Conclusions*

A tentative conclusion is made that there exists a correlation between agreement and accuracy, which may be used to determine confidence in the tagging of words by a group of taggers. The anomaly identified with respect to Table 6 could be attributed to a number of causes ranging from the taggers performing differently on different corpora (such as fiction), to the fact that no such relation between accuracy and agreement exists. Thus the proposed agreement metric needs to undergo more substantial testing on larger and more reliable corpora.

From the test on unrestricted data, confidence in groupings of taggers can now be tentatively measured for unannotated texts. In our case, as expected, the combination between the TreeTagger, Brill tagger and SVMTagger feature among the highest confidence levels.

4. CONCLUSION

This article discusses the early stages in a text-to-scene conversion system, and the importance of parts-of-speech taggers in such a system. In addition it gives a brief overview of the current state of the parts-of-speech tagging field of research, and presents preliminary results in applying combinations of such taggers to free text.

Contributions made by this paper include verifying the accuracies of various taggers on a different corpus. Various problems are identified in using a different corpus, specifically when translating tagsets. Additionally, a combination of taggers, which has not been tried before, has been tested using different voting schemes. Finally, a new metric is proposed to provide an indication of confidence in the accuracy of a group of taggers. Various forms of this metric are tested to determine a relationship with accuracy. A direct relationship is tentatively suggested, which requires more rigorous testing on larger corpora.

Future work includes evaluating combinations of the above taggers on the established test corpus, the Penn Treebank, as well as other corpora such as the Lancaster-Oslo/Bergen (LOB) corpus and the British National Corpus. These corpora, which are much larger than the SUSANNE corpus will provide a better indication of the relevance of the agreement metric, and its relation to accuracy. Additionally they have a much wider range of sources, and testing on such data will provide valuable insights regarding the choice of a tagger for use in a text-to-scene conversion system.

ACKNOWLEDGMENT

This work was undertaken in the Distributed Multimedia Centre of Excellence at Rhodes University, with financial support from Telkom SA, Business Connexion, Comverse, Verso Technologies, THRIP, and the National Research Foundation. The financial assistance from the Henderson Scholarship towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to Rhodes University or the donor.

REFERENCES

Brill, E. 1994. Some advances in transformation-based part of speech tagging. In *AAAI '94: Proceedings of the twelfth national conference on Artificial Intelligence (vol. 1)*. American Association for Artificial Intelligence, Menlo Park, CA, USA, 722–727.

Brill, E. and Wu, J. 1998. Classifier combination for improved lexical disambiguation. In *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, C. Boitet and P. Whitelock, Eds. Morgan Kaufmann Publishers, San Francisco, California, 191–195.

Coyne, B. and Sproat, R. 2001. Wordseye: an automatic text-to-scene conversion system. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press, 487–496.

Dimitrov, M. 2002. A light-weight approach to coreference resolution for named entites in text. M.S. thesis, University of Sofia.

Giménez, J. and Marquez, L. 2003. Fast and accurate part-of-speech tagging: The SVM approach revisited. In *RANLP*. 153–163.

Hobbs, J. R. 1978. Resolving pronoun references. *Lingua 44*, 311–338.

Kennedy, C. and Boguraev, B. 1996. Anaphora for everyone: Pronominal anaphora resolution without a parser. In *COLING*. 113–118.

Ma, M. E. 2002. Confucius: An intelligent multimedia storytelling interpretation and presentation system. Tech. rep., School of Computing and Intelligent Systems, University of Ulster, Magee. September.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. 1994. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics 19,* 2, 313–330.

Márquez, L., Rodríguez, H., Carmona, J., and Montolio, J. 1999. Improving POS tagging using machine-learning techniques. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. 53–62.

Mitkov, R., Evans, R., and Oruasan, C. 2002. A new, fully automatic version of Mitkov's knowledge-poor pronoun resolution method. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2002)*. Mexico City, Mexico.

Sampson, G. 2005. The SUSANNE Analytic Scheme, URL `http://www.grsampson.net/RSue.html`, [Accessed on 28 April 2005].

Schmid, H. 1994. Probabilistic part-of-speech tagging using decision trees. Tech. rep., IMS, Univ. of Stuttgart.

Schwartz, L. S. 1963. *Principles of coding, filtering, and information theory*. Spartan Books, Inc., Baltimore.

Toutanova, K. and Manning, C. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000)*. Hong Kong.

Toutanova, K., Manning, C., Klein, D., and Singer, Y. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL*. 252–259.

Tufis, D. and Mason, O. 1998. Tagging Romanian texts: a case study for QTag, a language independent probabilistic tagger. In *Proceedings First LREC*. Granada, Spain.

van Halteren, H., Zavrel, J., and Daelemans, W. 1998. Improving data driven wordclass tagging by system combination. In *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, C. Boitet and P. Whitelock, Eds. Morgan Kaufmann Publishers, San Francisco, California, 491–497.