

# Categorizing Unknown Words: Using Decision Trees to Identify Names and Misspellings

Janine Toole

Natural Language Laboratory  
Department of Computing Science  
Simon Fraser University  
Burnaby, BC, Canada V5A 1S6  
*toole@cs.sfu.ca*

## Abstract

This paper introduces a system for categorizing unknown words. The system is based on a multi-component architecture where each component is responsible for identifying one class of unknown words. The focus of this paper is the components that identify names and spelling errors. Each component uses a decision tree architecture to combine multiple types of evidence about the unknown word. The system is evaluated using data from live closed captions - a genre replete with a wide variety of unknown words.

## 1 Introduction

In any real world use, a Natural Language Processing (NLP) system will encounter words that are not in its lexicon, what we term 'unknown words'. Unknown words are problematic because a NLP system will perform well only if it recognizes the words that it is meant to analyze or translate: the more words a system does not recognize the more the system's performance will degrade. Even when unknown words are infrequent, they can have a disproportionate effect on system quality. For example, Min (1996) found that while only 0.6% of words in 300 e-mails were misspelled, this meant that 12% of the sentences contained an error (discussed in (Min and Wilson, 1998)).

Words may be unknown for many reasons: the word may be a proper name, a misspelling, an abbreviation, a number, a morphological variant of a known word (e.g. *recreated*), or missing from the dictionary. The first step in dealing with unknown words is to identify the class of the unknown word; whether it is a misspelling, a proper name, an abbreviation etc. Once this is known, the proper action can be taken, misspellings can be corrected, abbreviations can be expanded and so on, as deemed necessary by the particular text processing application. In this paper we introduce a system for categorizing unknown words. The system is based on a multi-component architecture where each component is responsible for identifying one category of unknown words. The main focus of this paper is the

components that identify names and spelling errors. Both components use a decision tree architecture to combine multiple types of evidence about the unknown word. Results from the two components are combined using a weighted voting procedure. The system is evaluated using data from live closed captions - a genre replete with a wide variety of unknown words.

This paper is organized as follows. In section 2 we outline the overall architecture of the unknown word categorizer. The name identifier and the misspelling identifier are introduced in section 3. Performance and evaluation issues are discussed in section 4. Section 5 considers portability issues. Section 6 compares the current system with relevant preceding research. Concluding comments can be found in section 6.

## 2 System Architecture

The goal of our research is to develop a system that automatically categorizes unknown words. According to our definition, an unknown word is a word that is not contained in the lexicon of an NLP system. As defined, 'unknown-ness' is a relative concept: a word that is known to one system may be unknown to another system.

Our research is motivated by the problems that we have experienced in translating live closed captions: live captions are produced under tight time constraints and contain many unknown words. Typically, the caption transcriber has a five second window to transcribe the broadcast dialogue. Because of the live nature of the broadcast, there is no opportunity to post-edit the transcript in any way. Although motivated by our specific requirements, the unknown word categorizer would benefit any NLP system that encounters unknown words of differing categories. Some immediately obvious domains where unknown words are frequent include e-mail messages, internet chat rooms, data typed in by call centre operators, etc.

To deal with these issues we propose a multi-component architecture where individual components specialize in identifying one particular type of

unknown word. For example, the misspelling identifier will specialize in identifying misspellings, the abbreviation component will specialize in identifying abbreviations, etc. Each component will return a confidence measure of the reliability of its prediction, c.f. (Elworthy, 1998). The results from each component are evaluated to determine the final category of the word.

There are several advantages to this approach. Firstly, the system can take advantage of existing research. For example, the name recognition module can make use of the considerable research that exists on name recognition, e.g. (McDonald, 1996), (Mani et al., 1996). Secondly, individual components can be replaced when improved models are available, without affecting other parts of the system. Thirdly, this approach is compatible with incorporating multiple components of the same type to improve performance (cf. (van Halteren et al., 1998) who found that combining the results of several part of speech taggers increased performance).

### 3 The Current System

In this paper we introduce a simplified version of the unknown word categorizer: one that contains just two components: misspelling identification and name identification. In this section we introduce these components and the 'decision' component which combines the results from the individual modules.

#### 3.1 The Name Identifier

The goal of the name identifier is to differentiate between those unknown words which are proper names, and those which are not. We define a name as word identifying a person, place, or concept that would typically require capitalization in English.

One of the motivations for the modular architecture introduced above, was to be able to leverage existing research. For example, ideally, we should be able to plug in an existing proper name recognizer and avoid the problem of creating our own. However, the domain in which we are currently operating - live closed captions - makes this approach difficult. Closed captions do not contain any case information, all captions are in upper case. Existing proper name recognizers rely heavily on case to identify names, hence they perform poorly on our data.

A second disadvantage of currently available name recognizers is that they do not generally return a confidence measure with their prediction. Some indication of confidence is required in the multi-component architecture we have implemented. However, while currently existing name recognizers are inappropriate for the needs of our domain, future name recognizers may well meet these requirements

and be able to be incorporated into the architecture we propose.

For these reasons we develop our own name identifier. We utilize a decision tree to model the characteristics of proper names. The advantage of decision trees is that they are highly explainable: one can readily understand the features that are affecting the analysis (Weiss and Indurkha, 1998). Furthermore, decision trees are well-suited for combining a wide variety of information.

For this project, we made use of the decision tree that is part of IBM's Intelligent Miner suite for data mining. Since the point of this paper is to describe an application of decision trees rather than to argue for a particular decision tree algorithm, we omit further details of the decision tree software. Similar results should be obtained by using other decision tree software. Indeed, the results we obtain could perhaps be improved by using more sophisticated decision-tree approaches such as the adaptive-resampling described in (Weiss et al, 1999).

The features that we use to train the decision tree are intended to capture the characteristics of names. We specify a total of ten features for each unknown word. These identify two features of the unknown word itself as well as two features for each of the two preceding and two following words.

The first feature represents the part of speech of the word. We use an in-house statistical tagger (based on (Church, 1988)) to tag the text in which the unknown word occurs. The tag set used is a simplified version of the tags used in the machine-readable version of the Oxford Advanced Learners Dictionary (OALD). The tag set contains just one tag to identify nouns.

The second feature provides more informative tagging for specific parts of speech (these are referred to as 'detailed tags' (DETAG)). This tagset consists of the nine tags listed in Table 1. All parts of speech apart from noun and punctuation tags are assigned the tag 'OTHER'. All punctuation tags are assigned the tag 'BOUNDARY'. Words identified as nouns are assigned one of the remaining tags depending on the information provided in the OALD (although the unknown word, by definition, will not appear in the OALD, the preceding and following words may well appear in the dictionary). If the word is identified in the OALD as a common noun it is assigned the tag 'COM'. If it is identified in the OALD as a proper name it is assigned the tag 'NAME'. If the word is specified as both a name and a common noun (e.g. 'bill'), then it is assigned the tag 'NCOM'. Pronouns are assigned the tag 'PRON'. If the word is in a list of titles that we have compiled, then the tag 'TITLE' is assigned. Similarly, if the word is a member of the class of words that can follow a name (e.g. 'jr'), then the tag 'POST' is assigned. A simple rule-based sys-

COM	common noun
NAME	name
NCOM	name and common noun
PRONOUN	pronoun
TITLE	title
POST	post-name word
BOUNDARY	boundary marker
OTHER	not noun or boundary
UNKNOWN	unknown noun

Table 1: List of Detailed Tags (DETAG)

Corpus frequency
Word length
Edit distance
Ispell information
Character sequence frequency
Non-English characters

Table 2: Features used in misspelling decision tree

tem is used to assign these tags.

If we were dealing with data that contains case information, we would also include fields representing the existence/non-existence of initial upper case for the five words. However, since our current data does not include case information we do not include these features.

### 3.2 The Misspelling Identifier

The goal of the misspelling identifier is to differentiate between those unknown words which are spelling errors and those which are not. We define a misspelling as an unintended, orthographically incorrect representation (with respect to the NLP system) of a word. A misspelling differs from the intended known word through one or more additions, deletions, substitutions, or reversals of letters, or the exclusion of punctuation such as hyphenation or spacing. Like the definition of ‘unknown word’, the definition of a misspelling is also relative to a particular NLP system.

Like the name identifier, we make use of a decision tree to capture the characteristics of misspellings. The features we use are derived from previous research, including our own previous research on misspelling identification. An abridged list of the features that are used in the training data is listed in Table 2 and discussed below.

*Corpus frequency:* (Vosse, 1992) differentiates between misspellings and neologisms (new words) in terms of their frequency. His algorithm classifies unknown words that appear infrequently as misspellings, and those that appear more frequently as neologisms. Our corpus frequency variable specifies the frequency of each unknown word in a 2.6 million

word corpus of business news closed captions.

*Word Length:* (Agirre et al., 1998) note that their predictions for the correct spelling of misspelled words are more accurate for words longer than four characters, and much less accurate for shorter words. This observation can also be found in (Kukich, 1992). Our word length variables measures the number of characters in each word.

*Edit distance:* Edit-distance is a metric for identifying the orthographic similarity of two words. Typically, one edit-distance corresponds to one substitution, deletion, reversal or addition of a character. (Damerau, 1964) observed that 80% of spelling errors in his data were just one edit-distance from the intended word. Similarly, (Mitton, 1987) found that 70% of his data was within one edit-distance from the intended word. Our edit distance feature represents the edit distance from the unknown word to the closest suggestion produced by the unix spell checker, ispell. If ispell does not produce any suggestions, an edit distance of thirty is assigned. In previous work we have experimented with more sophisticated distance measures. However, simple edit distance proved to be the most effective (Toole, 1999).

*Character sequence frequency:* A characteristic of some misspellings is that they contain character sequences which are not typical of the language, e.g. *tltd*, *wful*. Exploiting this information is a standard way of identifying spelling errors when using a dictionary is not desired or appropriate, e.g. (Hull and Srihari, 1982), (Zamora et al., 1981).

To calculate our character sequence feature, we firstly determine the frequencies of the two least frequent character tri-gram sequences in the word in each of a selection of corpora. In previous work we included each of these values as individual features. However, the resulting trees were quite unstable as one feature would be relevant to one tree, whereas a different character sequence feature would be relevant to another tree. To avoid this problem, we developed a composite feature that is the sum of all individual character sequence frequencies.

*Non-English characters:* This binary feature specifies whether a word contains a character that is not typical of English words, such as accented characters, etc. Such characters are indicative of foreign names or transmission noise (in the case of captions) rather than misspellings.

### 3.3 Decision Making Component

The misspelling identifier and the name identifier will each return a prediction for an unknown word. In cases where the predictions are compatible, e.g. where the name identifier predicts that it is a name and the spelling identifier predicts that it is not a misspelling, then the decision is straightforward. Similarly, if both decision trees make negative predictions, then we can assume that the unknown word

is neither a misspelling nor a name, but some other category of unknown word.

However, it is also possible that both the spelling identifier and the name identifier will make positive predictions. In these cases we need a mechanism to decide which assignment is upheld. For the purposes of this paper, we make use of a simple heuristic where in the case of two positive predictions the one with the highest confidence measure is accepted. The decision trees return a confidence measure for each leaf of the tree. The confidence measure for a particular leaf is calculated from the training data and corresponds to the proportion of correct predictions over the total number of predictions at this leaf.

## 4 Evaluation

In this section we evaluate the unknown word categorizer introduced above. We begin by describing the training and test data. Following this, we evaluate the individual components and finally, we evaluate the decision making component.

The training and test data for the decision tree consists of 7000 cases of unknown words extracted from a 2.6 million word corpus of live business news captions. Of the 7000 cases, 70.4% were manually identified as names and 21.3% were identified as misspellings. The remaining cases were other types of unknown words such as abbreviations, morphological variants, etc. Seventy percent of the data was randomly selected to serve as the training corpus. The remaining thirty percent, or 2100 records, was reserved as the test corpus. The test data consists of ten samples of 2100 records selected randomly with replacement from the test corpus.

We now consider the results of training a decision tree to identify misspellings using those features we introduced in the section on the misspelling identifier. The tree was trained on the training data described above. The tree was evaluated using each of the ten test data sets. The average precision and recall data for the ten test sets are given in Table 3, together with the base-line case of assuming that we categorize all unknown words as names (the most common category). With the baseline case we achieve 70.4% precision but with 0% recall. In contrast, the decision tree approach obtains 77.1% precision and 73.8% recall.

We also trained a decision tree using not only the features identified in our discussion on misspellings but also those features that we introduced in our discussion of name identification. The results for this tree can be found in the second line of Table 3. The inclusion of the additional features has increased precision by approximately 5%. However, it has also decreased recall by about the same amount. The overall F-score is quite similar. It appears that

the name features are not predictive for identifying misspellings in this domain. This is not surprising considering that eight of the ten features specified for name identification are concerned with features of the two preceding and two following words. Such word-external information is of little use in identifying a misspelling.

An analysis of the cases where the misspelling decision tree failed to identify a misspelling revealed two major classes of omissions. The first class contains a collection of words which have typical characteristics of English words, but differ from the intended word by the addition or deletion of a syllable. Words in this class include *creditability* for *credibility*, *coordinatedored* for *coordinated*, and *representives* for *representatives*. The second class contains misspellings that differ from known words by the deletion of a blank. Examples in this class include *webpage*, *crewmembers*, and *rainshower*. The second class of misspellings can be addressed by adding a feature that specifies whether the unknown word can be split up into two component known words. Such a feature should provide strong predictability for the second class of words. The first class of words are more of a challenge. These words have a close homophonic relationship with the intended word rather than a close homographic relationship (as captured by edit distance). Perhaps this class of words would benefit from a feature representing phonetic distance rather than edit distance.

Among those words which were incorrectly identified as misspellings, it is also possible to identify common causes for the misidentification. Among these words are many foreign words which have character sequences which are not common in English. Examples include *khanchanalak*, *phytoplankton*, *bryceen*.

The results for our name identifier are given in Table 4. Again, the decision tree approach is a significant improvement over the baseline case. If we take the baseline approach and assume that all unknown words are names, then we would achieve a precision of 70.4%. However, using the decision tree approach, we obtain 86.5% precision and 92.9% recall.

We also trained a tree using both the name and misspelling features. The results can be found in the second line of Table 4. Unlike the case when we trained the misspelling identifier on all the features, the extended tree for the name identifier provides increased recall as well as increased precision. Unlike the case with the misspelling decision-tree, the misspelling-identification features do provide predictive information for name identification. If we review the features, this result seems quite reasonable: features such as corpus frequency and non-English characters can provide evidence for/against name iden-

	Baseline Precision/Recall	Precision	Recall	F-score
Misspelling features only	70.4%/0%	73.8%	77.1%	75.4
All features		82.8%	68.9%	75.2

Table 3: Precision and recall for misspelling identification

tification as well as for/against misspelling identification. For example, an unknown word that occurs quite frequently (such as *clinton*) is likely to be a name, whereas an unknown word that occurs infrequently (such as *wful*) is likely to be a misspelling.

A review of the errors made by the name identifier again provides insight for future development. Among those unknown words that are names but which were not identified as such are predominantly names that can (and did) appear with determiners. Examples of this class include *steelers* in *the steelers*, and *pathfinder* in *the pathfinder*. Hence, the name identifier seems adept at finding the names of individual people and places, which typically cannot be combined with determiners. But, the name identifier has more problems with names that have similar distributions to common nouns.

The cases where the name identifier incorrectly identifies unknown words as names also have identifiable characteristics. These examples mostly include words with unusual character sequences such as the misspellings *szetion* and *fulamg*. No doubt these have similar characteristics to foreign names. As the misidentified words are also correctly identified as misspellings by the misspelling identifier, these are less problematic. It is the task of the decision-making component to resolve issues such as these.

The final results we include are for the unknown word categorizer itself using the voting procedure outlined in previous discussion. As introduced previously, confidence measure is used as a tie-breaker in cases where the two components make positive decision. We evaluate the categorizer using precision and recall metrics. The precision metric identifies the number of correct misspelling or name categorizations over the total number of times a word was identified as a misspelling or a name. The recall metric identifies the number of times the system correctly identifies a misspelling or name over the number of misspellings and names existing in the data. As illustrated in Table 5, the unknown word categorizer achieves 86% precision and 89.9% recall on the task of identifying names and misspellings.

An examination of the confusion matrix of the tie-breaker decisions is also revealing. We include the confusion matrix for one test data set in Table 6. Firstly, in only about 5% of the cases was it necessary to revert to confidence measure to determine the category of the unknown word. In all other cases the predictions were compatible. Secondly, in the

majority of cases the decision-maker rules in favour of the name prediction. In hindsight this is not surprising since the name decision tree has higher results and hence is likely to have higher confidence measures.

A review of the largest error category in this confusion matrix is also insightful. These are cases where the decision-maker classifies the unknown word as a name when it should be a misspelling (37 cases). The words in this category are typically examples where the misspelled word has a phonetic relationship with the intended word. For example, *temt* for *tempt*, *floyda* for *florida*, and *dimow* part of the intended word *democrat*. Not surprisingly, it was these types of words which were identified as problematic for the current misspelling identifier. Augmenting the misspelling identifier with features to identify these types of misspellings should also lead to improvement in the decision-maker.

We find these results encouraging: they indicate that the approach we are taking is productive. Our future work will focus on three fronts. Firstly, we will improve our existing components by developing further features which are sensitive to the distinction between names and misspellings. The discussion in this section has indicated several promising directions. Secondly, we will develop components to identify the remaining types of unknown words, such as abbreviations, morphological variants, etc. Thirdly, we will experiment with alternative decision-making processes.

## 5 Examining Portability

In this paper we have introduced a means for identifying names and misspellings from among other types of unknown words and have illustrated the process using the domain of closed captions. Although not explicitly specified, one of the goals of the research has been to develop an approach that will be portable to new domains and languages.

We are optimistic that the approach we have developed is portable. The system that we have developed requires very little in terms of linguistic resources. Apart from a corpus of the new domain and language, the only other requirements are some means of generating spelling suggestions (ispell is available for many languages) and a part-of-speech tagger. For this reason, the unknown word categorizer should be portable to new languages, even where extensive language resources do not exist. If

	Baseline Precision	Precision	Recall	F-score
Name features only	70.4%	86.5%	92.9%	89.6
All Features		91.8%	94.5%	93.1

Table 4: Precision and recall for name identification

	Precision	Recall	F-score
Predicting Names and Misspellings	86.6%	89.9%	88.2

Table 5: Precision and recall for decision-making component

more information sources are available, then these can be readily included in the information provided to the decision tree training algorithm.

For many languages, the features used in the unknown word categorizer may well be sufficient. However, the features used do make some assumptions about the nature of the writing system used. For example, the edit distance feature in the misspelling identifier assumes that words consist of alphabetic characters which have undergone substitution/addition/deletion. However, this feature will be less useful in a language such as Japanese or Chinese which use ideographic characters. However, while the exact features used in this paper may be inappropriate for a given language, we believe the general approach is transferable. In the case of a language such as Japanese, one would consider the means by which misspellings differ from their intended word and identify features to capture these differences.

## 6 Related Research

There is little research that has focused on differentiating the different types of unknown words. For example, research on spelling error detection and correction for the most part assumes that all unknown words are misspellings and makes no attempt to identify other types of unknown words, e.g. (Elmi and Evens, 1998). Naturally, these are not appropriate comparisons for the work reported here. However, as is evident from the discussion above, previous spelling research does provide an important role in suggesting productive features to include in the decision tree.

Research that is more similar in goal to that outlined in this paper is Vosse (Vosse, 1992). Vosse uses a simple algorithm to identify three classes of unknown words: misspellings, neologisms, and names. Capitalization is his sole means of identifying names. However, capitalization information is not available in closed captions. Hence, his system would be ineffective on the closed caption domain with which we are working. (Granger, 1983) uses expectations generated by scripts to analyze unknown words. The drawback of his system is that it lacks portability

since it incorporates scripts that make use of world knowledge of the situation being described; in this case, naval ship-to-shore messages.

Research that is similar in technique to that reported here is (Baluja et al., 1999). Baluja and his colleagues use a decision tree classifier to identify proper names in text. They incorporate three types of features: word level (essentially utilizes case information), dictionary-level (comparable to our ispell feature), and POS information (comparable to our POS tagging). Their highest F-score for name identification is 95.2, slightly higher than our name identifier. However, it is difficult to compare the two sets of results since our tasks are slightly different. The goal of Baluja's research, and all other proper name identification research, is to identify all those words and phrases in the text which are proper names. Our research, on the other hand, is not concerned with all text, but only those words which are unknown. Also preventing comparison is the type of data that we deal with. Baluja's data contains case information whereas ours does not- the lack of case information makes name identification significantly more difficult. Indeed, Baluja's results when they exclude their word-level (case) features are significantly lower: a maximum F-score of 79.7.

## 7 Conclusion

In this paper we have introduced an unknown word categorizer that can identify misspellings and names. The unknown word categorizer consists of individual components, each of which specialize in identifying a particular class of unknown word. The two existing components are implemented as decision trees. The system provides encouraging results when evaluated against a particularly challenging domain: transcripts from live closed captions.

## References

- E. Agirre, K. Gojenola, K. Sarasola, , and A. Voutilainen. 1998. Towards a single proposal in spelling correction. In *Proceedings of the 36th Annual Meeting of the ACL and the 17th International*

	Predicted Spelling	Predicted Name
Neither name nor misspelling	0	6
Misspelling	10	37
Name	4	43

Table 6: Confusion matrix for decision maker: includes only those examples where both components made a positive prediction.

- Conference on Computational Linguistics*, pages 22–28.
- S. Baluja, V. Mittal, and R. Sukthankar. 1999. Applying machine learning for high performance named-entity extraction. In *Proceedings of the Conference of the Pacific Association for Computational Linguistics*, pages 365–378.
- K. Church 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pages 136–143.
- F. Damerau. 1964. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7:171–176.
- M. Elmi and M. Evens. 1998. Spelling correction using context. In *Proceedings of the 36th Annual Meeting of the ACL and the 17th International Conference on Computational Linguistics*, pages 360–364.
- D. Elworthy. 1998. Language identification with confidence limits. In *Proceedings of the 6th Workshop on Very Large Corpora*.
- R. Granger. 1983. The nomad system: expectation-based detection and correction of errors during understanding of syntactically and semantically ill-formed text. *American Journal of Computational Linguistics*, 9:188–198.
- J. Hull and S. Srihari. 1982. Experiments in text recognition with binary n-gram and viterbi algorithms. *IEEE Trans. Patt. Anal. Machine Intell. PAMI-4*, 5:520–530.
- K. Kukich. 1992. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24:377–439.
- I. Mani, R. McMillan, S. Luperfoy, E. Lusher, and S. Laskowski, 1996. *Corpus Processing for Lexical Acquisition*, chapter Identifying unknown proper names in newswire text. MIT Press, Cambridge.
- D. McDonald, 1996. *Corpus Processing for Lexical Acquisition*, chapter Internal and external evidence in the identification and semantic categorization of proper names. MIT Press, Cambridge.
- K. Min and W. Wilson. 1998. Integrated control of chart items for error repair. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*.
- K. Min. 1996. *Hierarchical Error Recovery Based on Bidirectional Chart Parsing Techniques*. Ph.D. thesis, University of NSW, Sydney, Australia.
- R. Mitton. 1987. Spelling checkers, spelling correctors, and the misspellings of poor spellers. *Inf. Process. Manage*, 23:495–505.
- J. Toole 1999 Categorizing Unknown Words: A decision tree-based misspelling identifier In Foo, N (ed.) *Advanced Topics in Artificial Intelligence*, pages 122–133.
- H. van Halteren, J. Zavrel, and W. Daelemans. 1998. Improving data driven word class tagging by system combination. In *Proceedings of the 36th Annual Meeting of the ACL and the 17th International Conference on Computational Linguistics*, pages 491–497.
- T. Vosse. 1992. Detecting and correcting morphosyntactic errors in real texts. In *Proceedings of the 3rd Conference on Applied Natural Language Processing*, pages 111–118.
- S. Weiss and N. Indurkha. 1998. *Predictive Data Mining*. Morgan Kauffman Publishers.
- S. Weiss, and C. Apte, and F. Damerau, and D. Johnson, and F. Oles and T. Goetz, and T. Hampp. 1999 Maximizing text-mining performance. *IEEE Intelligent Systems and their Applications*, 14(4):63–69
- E. Zamora, J. Pollock, and A. Zamora. 1981. The use of tri-gram analysis for spelling error detection. *Inf. Process. Manage.*, 17:305–316.