CONTENT CHARACTERIZATION USING WORD SHAPE TOKENS

Penelope Sibun and David S. Farrar
Fuji Xerox Palo Alto Laboratory, 3400 Hillview Avenue, Palo Alto, CA 94304
sibun@pal.xerox.com, farrar@pal.xerox.com

Abstract

By quickly classifying character images into character shape categories, it is possible to automatically extract syntactic information from the text of document images without optical character recognition. Using *word shape tokens* composed of these *character shape codes*, a properly trained text tagger can extract part-of-speech information from scanned document images. Later components of a document processing system can then use this information to locate topics, characterize document style, and assist in information retrieval.

1 INTRODUCTION

There are many text processing tasks that we would like to accomplish, such as document classification, text database structuring, matching documents with queries, and topic characterization. The field of computational linguistics has developed a variety of techniques for accomplishing these tasks for text documents represented by character codes (e.g., ASCII). However, many documents for which we would like to use our automated techniques are not stored online in character-coded format, but instead exist only on paper. Optical character recognition (OCR) is a technique for converting scanned document images into character codes. By using OCR, document images can be converted into a form amenable to existing text processing techniques. However, OCR is expensive, slow, and often inaccurate. Because of these drawbacks, we would like to avoid OCR if we can, or at the least, postpone using OCR until we are confident that a document warrants detailed processing. In other words, we would like a high-bandwidth document processing system that is sensitive enough to detect desired document features.

Our document understanding goals at the Fuji Xerox Palo Alto Laboratory include *language determination* (Nakayama and Spitz, 1993; Sibun and Spitz, forthcoming), *content characterization*, and *style characterization*. Toward these goals, we are developing a set of methods for extracting information from document images which do not depend on OCR. We have been working toward our goal of inexpensive content characterization by adapting a *part-of-speech tagger* to process word shape tokens rather than character coded words. Part-of-speech tagging is a technique that has been developed and refined over the past several years, and it provides an inexpensive, fast, and reliable source of information for recognizing noun phrases and other syntax-related text features which help characterize a document's content.

In this paper, we describe how we combine our technology for determining word shape tokens with text-tagging technology. We are developing systems that can extract noun phrases and other content characteristics using only word shape tokens that have been tagged with their parts of speech. Using this approach, we can process document images quickly to determine whether OCR is warranted, for example, when a text is a likely match for keywords in a database query.

In the next two sections, we describe how word shape tokens are derived; in section four, we discuss part-of-speech tagging; in the following four sections, we describe in detail part-of-speech tagging using word shape tokens; in sections nine and ten we discuss our results.

2 WORD SHAPE TOKEN CREATION

In this section we briefly describe our system that constructs character shape codes and word shape tokens from a document image (for more detail, see Nakayama and Spitz, 1993; Sibun and Spitz, forthcoming). To recognize *character shape codes* from an image, some transformations are first made to correct for various scanning artifacts such as skew angle and text line curvature. On each text line, four horizontal lines define three significant zones: the area between the baseline and the top of characters such as "x" is the *x-zone*; the area above the x-height level is the *ascender zone*; the area below the x-zone is the *descender zone* (figure 1). The text line is further divided into *character cells* by vertical boundaries which delineate the connected components of each character image.

Figure 1: The text line parameter positions.

The majority of characters can easily be mapped to a small number of distinct codes (figure 2).[1] Characters which are contained entirely in the x-zone map to shape code **x**; characters which extend from the baseline to above the x-height line map to shape code **A**; and those which extend from below the baseline to the x-height line map to shape code **g**. Characters which map to **A**, **x**, or **g** are composed of a single connected component. Some characters contain more than one connected component: an x-height character with a single diacritical mark in the ascender zone maps to **i**; a character with a descender and a single diacritical mark maps to **j**. Most common punctuation marks map to unique shape codes; however,

[1] If this mapping can be done from document images, it can more trivially be accomplished from character-coded documents, such as ASCII text (providing, of course, that the method of encoding is known).

some are mapped into shape codes shared with alphabetic characters (e.g., "&" maps to shape code **A**).

Shape Code	Character
A	A - Z b d f h k l t 0 - 9 # $ & / @ |
x	a c e m n o r s u v w x z
i	ì á à â é è ê í ì î ó ò ô ú ù û ñ
g	g p q y ç
j	j

Figure 2: Character shape codes.

3 SHAPE CONVERSION

In general, our approach to document processing finesses the problems inherent in mapping from an image to a character-coded representation: we map instead from the image to a *shape-based representation*. This technique can transform even a degraded document image into a representation which provides useful abstractions about the text of a document. The shape-based representation that we construct is proving to be a remarkably rich source of information. While our initial goal has been to use it for language identification in support of downstream OCR processes, we are finding that this representation may be a sufficient source of information for document content characterization, such as that supported by part-of-speech tagging.

In our tagging work, we have used character shape coded text derived from normal character-coded text. This is simply because we do not have access to enough image documents on which to train a tagger. We call the process of creating a shape-based version of the document from the character code based version *shape conversion*.

For the purpose of text tagging, then, we can think of the word shape token representation as an approximation of the representation composed of words. We can think about the relationship between words and word shape tokens as a mapping from a word to its corresponding word shape token. For example, the word "apple" maps to the word shape token **xggAx**, and the word "apples" maps to the word shape token **xggAxx**.

In documents, words exist as *surface forms*, not as morphological systems; thus "apple" and "apples" are different words. Therefore, it is of no use to us to have a lexicon organized in terms of stems and suffixes; instead, our lexicon is composed of surface forms like "apple" and "apples". Throughout the rest of this paper, when we say "words", we mean words as surface forms.

4 PART-OF-SPEECH TAGGING

A *part-of-speech tagger* is a system that uses context to assign parts of speech to words. Part-of-speech information facilitates higher-level analysis, such as recognizing noun phrases and other patterns in text. Several different approaches have been used for building text taggers. A particular form of Markov model has been widely used that assumes that a word depends probabilistically on just its part-of-speech category, which in turn depends solely on the categories of the preceding two words. Training the model is sometimes done by means of a large tagged corpus, but this is not necessary. The Baum-Welch algorithm (Baum, 1972), also known as the Forward-Backward algorithm, can be used. In this case, the model is called a *hidden Markov model* (HMM), since state transitions (i.e., part-of-speech categories) are assumed to be unobservable.

For this work, we use an HMM-based text tagger that is publicly available from Xerox PARC. As described in Cutting et al. (1992), the PARC tagger is efficient and highly flexible. It is particularly important that the tagger can be trained on any corpus of text, using any lexicon. This flexibility allows us to shape-convert our training corpus and lexicon, as described in section 5, without needing to modify the tagger itself. Below we outline the basic operation of the PARC tagger; please refer to Cutting et al. (1992) for further detail.

1. Text destined for the tagger first encounters a tokenizer, whose duty is to convert text (a sequence of characters) into a sequence of tokens. Each sentence boundary is also identified by the tokenizer, and is passed as a special token.

2. The tokenizer passes tokens to the lexicon, where tokens are matched with a set of surface forms, each annotated with a part-of-speech tag. The set of tags constitutes an *ambiguity class*. The lexicon passes along a stream of (*surface form*, *ambiguity class*) pairs.

3a. In training mode, the tagger takes long sequences of ambiguity classes as input. It uses the Baum-Welch algorithm to produce a trained HMM, which is used as input in tagging mode. Training is performed on some corpus of interest; this corpus may be of broad coverage or may be genre-specific.

3b. In tagging mode, the tagger buffers sequences of ambiguity classes between sentence boundaries. These sequences are disambiguated by computing the maximal path through the HMM with the Viterbi algorithm (1967). Operating at sentence granularity does not sacrifice accuracy, since sentence boundaries are unambiguous. Output consists of pairs of surface forms and tags.

5 THE LEXICON

The word shape tagging in our work follows the HMM-based process described above. Both word shape tagging and standard word tagging require a lexicon.

5.1 Constructing the Lexicon

A word shape lexicon can be derived from a standard lexicon of words. The lexicon used with the standard text tagger contains a list of all the distinct surface forms likely to be encountered in the language. Associated with each surface form is a list of the possible parts of speech that the surface form can have. For example:

apple	noun
apples	plural noun
eat	verb
eats	third person singular verb
red	noun, adjective
the	determiner

Once we have a lexicon which consists of surface forms, we can use it to create a lexicon of word shape tokens for

word shape tagging. In particular, this transformation consists of the following steps:

1. Shape convert the surface forms to their corresponding word shape tokens.
2. Sort the lexicon by surface form word shape. At this stage there may be duplicate word shape tokens.
3. Eliminate duplicate entries in the lexicon: collect all parts of speech behind one word shape token (combine their ambiguity classes). At this stage each word shape token should be unique.
4. Eliminate duplicate parts of speech behind each word shape token. At this stage each part of speech should be unique within each ambiguity class.

The lexicon fragment above would be converted to:

xggAx	noun
xggAxx	plural noun
xxA	verb, noun, adjective
xxAx	third person singular verb
AAx	determiner

5.2 Analysis of the Lexicon

For this work, we use a lexicon provided by Xerox PARC. This lexicon is organized so that there is an entry for each of roughly 150,000 surface forms. For word shape tagging, we shape converted this lexicon. As can be seen in the table, shape conversion results in about 50,000 *distinct* word shape surface forms. This suggests that, on average, each word shape token is a mapping of three surface forms. However, about 30,000 of the word shape tokens are *unique*, that is, correspond to a single surface form.

Surface Forms	Count	%Total
Standard Lexicon	148,703	100.0
Shape-converted Lexicon	47,102	31.7
Shape-converted Unique	28,949	19.5

Thus, the word shape lexicon is approximately one-third the size of the standard lexicon. Clearly, information has been lost, but not as much as one might think. In fact, the 20% of the word shape tokens that are unique carry exactly as much information as their corresponding character-coded words. While some surface forms that map to unique word shape tokens are long and infrequent (like "flibbertigibbet", **AAiAAxxAigiAAxA**), many are short, common words:

apple	**xggAx**
apples	**xggAxx**
thigh	**AAigA**
thirsty	**AAixxAg**
lifelike	**AiAxAiAx**
galaxy	**gxAxxg**
payday	**gxgAxg**
paydays	**gxgAxgx**

While word shape tokens that are unique have the same parts of speech as their corresponding surface forms, the others will tend on average to have many more parts of speech than an average surface form. This depends somewhat on the tagset (see section 6). In general, word shape tokens frequently have as many as 10 to 15 parts of speech, whereas standard surface forms rarely have more than 4 or 5.

6 DEVISING THE TAGSET

The *tagset* is implicit in the lexicon: it includes all parts of speech listed in any entry of the lexicon; it also includes a small set of tags for punctuation, such as comma, hyphen, and sentence boundary. Although the tagset is not explicitly defined, we can modify it by mapping from selected tags found in the lexicon to other tags of our choosing. For example, the lexicon distinguishes between verb tenses and has separate tags for different combinations of verb tense, person, and number: *present tense verb*, *past tense verb*, *third person singular present verb*, etc. If we preferred, we could map all these different verb forms to a single *verb* tag. However, we typically prefer to maintain such distinctions, as the text tagger can take advantage of differences in the surface forms of verbs with different tenses in order to uniquely identify their parts of speech.

Shape conversion collapses different surface forms onto one word shape and merges their ambiguity classes. The result is that there tend to be fewer distinct surface forms, and that each surface form has, on average, a larger ambiguity class. If this ambiguity is problematic, one way to reduce it may be to reduce the size of the tagset. For example, we may choose to have one undifferentiated verb tag rather than a set which differentiates tense, person, and number. With fewer possible parts of speech to choose from, the HMM may find the part-of-speech selection more constrained. This in turn may improve its accuracy at selecting one of the tags that are available.

The uninteresting case, of course, is where every word shape has the same tag, that is, a tag set of one. This situation yields no useful syntactic information from the document. Since the use of word shape tokens does reduce the amount of information that is available to the tagger, it may reduce the number of different tags it can accurately assign. The proper size of the tagset becomes constrained on one hand by the amount of syntactic information we wish to extract (more information with a larger tagset) and on the other by the size of the ambiguity classes of the word shape tokens (more ambiguity with a larger tagset). Its proper size is thus an empirical question. For our tests we used tagsets with approximately 30 parts of speech.

7 THE TRAINING PROCESS

Just as the hidden Markov model for standard text tagging requires a large corpus of text to train on, the word shape HMM requires a large corpus of text that has been converted to word shape tokens. We used at least 3.5 megabytes of ASCII text for our standard text tagger's corpus; we then shape converted this text to create the corpus for the word shape tagger. This corpus consisted of a variety of different writing styles (from colloquial to professional) and difficulty levels (from casual to erudite). Examples include essays by humorists, proposals for new government policies, and classic works of literature.

8 THE TAGGING PROCESS

With the word shape lexicon in place and an adequately trained HMM, word shape tagging works just as standard text tagging does. In particular, word shape tagging consists of the following steps:

1. A stream of text is tokenized into a stream of word shape tokens segmented into sentences.

2. The shape-converted lexicon assigns an ambiguity class to each word shape token. The result is a stream of sentences composed of (*word shape token*, *ambiguity class*) pairs.

3. The tagger uses the trained hidden Markov model to compute the highest probability part of speech for each word shape token in a sentence. The result is a stream of (*word shape token*, *part of speech*) pairs, which are grouped according to sentence boundaries.

We can now use the resulting parts of speech to inform other segments of a document understanding system. The word shape part-of-speech tagger thus accepts word shape tokens grouped by sentence boundaries; within those boundaries, it assigns the most likely part of speech to each word shape token.

9 RESULTS

In this section, we introduce a tool which can recognize noun phrases in sentences, and we use this tool to compare the performance of the standard tagger and the word shape tagger. We exemplify the comparison with two texts: one on which the standard tagger performs very well, and one on which it does relatively poorly. While the word shape tagger does less well in each case, its behavior tracks that of the standard tagger, exhibiting similar successes and failures. For the particular task of finding simple noun phrases, the word shape tagger's performance is less than that of the standard tagger's, but a large fraction of the noun phrases still are found.

We have a system that can recognize simple noun phrases when given as input the sequence of tags for a sentence. Each of these phrases comprises a contiguous sequence of tags that satisfies a simple grammar. For example, a noun phrase can be simply a pronoun tag or an arbitrary sequence of noun and adjective tags, possibly preceded by a determiner tag and possibly with an embedded possessive tag.[2] The longest possible such sequences are found. Conjunctions are not recognized as part of a noun phrase, nor is prepositional phrase attachment performed. We can be confident of finding many simple noun phrases because the word "the" has the unique word shape **AAx**.[3] Recognition of noun phrases is a first step in topic identification: the topic of a document is likely to be indicated by its most frequent noun phrases.

In evaluating the tagger error rate, we use several measures (see tables). We calculate the percentage of *total errors*, the percentage of *trivial errors*, and the percentage of *pernicious errors* (there are a few errors that do not fall in either of the latter categories). Tagging "alarming" in "what the advocates are finding alarming" as a present participle rather than as an adjective is an example of a trivial error. Pernicious errors typically involve mistagging nouns as verbs or verbs as nouns (in English, there are many surface forms that can be either nominal or verbal). These latter errors cause problems in later processing, such as detecting simple noun phrases, since they may obscure noun phrases or introduce spurious ones.

We compare the standard tagger and the word shape tagger by counting the *matches* in the streams of output tags. We do not demand strict matches, but instead allow the tags to belong to pertinent equivalence classes. For example, the standard tagger labels the noun "monitors" as a plural noun, and the word shape tagger labels **xxxiAxxx** simply as a noun. We consider this a match, since a noun and a plural noun are equally well recognized as part of a noun phrase.

Almost all instances of mismatches result from the standard tagger being right and the word shape tagger being wrong. Very occasionally the situation is the reverse, but this is to be expected as within the normal range of probabilities. More interesting is the observation that almost every pernicious error made by the standard tagger is repeated by the word shape tagger. We take this as confirmation of the word shape tagger's ability to approximate the standard tagger's performance.

The first comparison of tagger performance involves a 394-word excerpt from a government document. The standard tagger's performance is better than 95% correct, or better than 97% if trivial errors are disregarded. The word shape tagger's performance is a 59% match of the standard tagger's (or 51% if only exact matches are considered). The noun phrase recognizer found 113 simple noun phrases in the standard tagger's output and 77 (68%) of these in the word shape tagger's output.

Standard Tagger Errors

Text	Total	Trivial	Pernicious	Other
Government	4.6%	2.0%	2.3%	0.5%
Nonsense	11.1%	4.2%	4.9%	2.0%

Matching Output of Standard Tagger and Word Shape Tagger

Text	Disregarding Trivial Mismatches	Including all Mismatches
Government	59%	51%
Nonsense	47%	38%

Noun Phrases Recognized from Tagger Output

Text	Standard	Word Shape
Government	113	77
Nonsense	45	17

The second comparison is of tagging a 144-word piece of nonsense verse. The standard tagger's performance is

[2] The possessive tag is used for "'s" or "s'" as in "the cat's pajamas' stripes"

[3] Another English word, "flu," also maps to **AAx**; fortunately, in most contexts this word is rare.

89% correct, or 94% disregarding trivial errors. The word shape tagger's performance is a 47% match (or 38% considering only exact matches). The noun phrase recognizer found 45 simple noun phrases in the standard tagger's output and 17 (38%) of these in the word shape tagger's output.

Further study is needed to determine exactly how reliable word shape part-of-speech tagging and simple noun phrase recognition will be in finding the topic or topics in a document image. One means of improving this reliability may be our technique for grammatical function assignment which uses only the output of the part-of-speech tagger and phrase recognizers (Sibun 1991). However, we can already use part-of-speech tagging and simple noun phrase recognition as a tool for discerning something about the content of the document by discovering at least some of its noun phrases. Since our document recognition technology allows us to use word shape tokens to index directly into the document image, we can also identify parts of the image as promising candidates for OCR.

10 DISCUSSION

Although the word shape tagger deals with greater ambiguity, it can still extract significant information from a text. The increase in ambiguity is not as high as might be expected: a large number of word shapes remain unambiguous after the lexicon has been shape converted. As noted above, the creation of the word shape lexicon from the standard lexicon reduces the number of distinct entries to approximately one-third. For example, distinct words such as "cat" and "rat" map onto the same word shape token **xxA**. Nevertheless, the complexity of English spelling still allows a large proportion of surface forms to be distinguished merely by their word shapes.

Several improvements on our technique remain to be fully implemented. We do not yet have a principled way to determine the optimal tagset for a given corpus of text. As noted above, there is a tension between the size of the tagset and the amount of syntactic information that is available in the word shape tokens.

We are also investigating computationally inexpensive ways of making finer distinctions between characters that map to the character shape codes **x** and **A**. Initially, parentheses and brackets were always classified as **A** and distorted any word shape they were adjacent to: for example, "(USA)" would be shape converted to **AAAAA**. Recently we have made progress in recognizing these non-alphabetic characters as word shape token delimiters, rather than parts of the word shape tokens themselves. It may also be useful to distinguish more alphabetic character classes by mapping scanned character images to a larger set of character shape codes. We can extract more useful information by distinguishing upper case letters from lower case letters, such as "h" and "k", which map to the character shape code **A**. A larger number of character shape codes gives us more information about the word shape tokens, and helps to reduce ambiguity. However, we must be careful to choose character shape features which can be easily detected in the image and quickly classified by a character shape code.

In keeping with Fuji Xerox's multi-lingual document emphasis, we are also exploring ways in which this method may be applied to other Roman-alphabet languages, such as French, German, Dutch, and Spanish. The technique will need to be evaluated separately for each language, however, to better understand how each language's typographic conventions may be reflected in its word shape.

11 CONCLUSION

We have presented a new technique for the understanding of English document images without optical character recognition. By scanning and categorizing character shapes, it is possible to extract word shapes from the document text; these word shapes tokens can then be used as input to a tagger which determines part-of-speech information. This part-of-speech information can then be used to inform other document understanding techniques, including noun phrase recognition and topic identification. The lack of OCR means we cannot extract all of the information contained in the scanned document's image; nevertheless, the information from the word shape tokens allows us to characterize the document's content with significant accuracy, and more quickly than if we had performed OCR.

Acknowledgments

We thank Larry Spitz and Masa Ozaki for their useful comments.

References

Baum, L. E. "An inequality and associated maximization technique in statistical estimation for probabilistic functions of a Markov process." *Inequalities*, 3:1--8, 1972.

Cutting, Doug, Julian Kupiec, Jan Pedersen, and Penelope Sibun. "A Practical Part-of-Speech Tagger." In *Proceedings of the Third Conference on Applied Natural Language Processing (ACL)*, pp 133-140, Trento, Italy, 1992. Also Report SSL-92-01/P92-00001, Xerox Palo Alto Research Center, 1992.

Nakayama, Takehiro and A. L. Spitz. "European Language Determination from Image." In Proceedings of the Second International Conference on Document Analysis and Recognition, pp 159-162, Tsukuba Science City, Japan, 1993.

Sibun, Penelope. "Grammatical Function Assignment in Unrestricted Text." Internal Report, Xerox Palo Alto Research Center, 1991.

Sibun, Penelope and A. Lawrence Spitz. "Language Determination: Natural Language Processing from Scanned Document Images." Forthcoming.

Viterbi, A. J. "Error bounds for convolution codes and an asymptotically optimal decoding algorithm." *IEEE Transactions on Information Theory*. pp 260-269. April 1967.