From Pixels to Features, J.C. Simon (ed.)

FEATURE SELECTION AND LANGUAGE SYNTAX IN TEXT RECOGNITION

Jonathan J. Hull

Department of Computer Science
State University of New York at Buffalo
Buffalo, New York 14260

There are many features that can be used to recognize images of text. The choice of a feature set is usually made intuitively to optimize performance in single character recognition. This approach to feature set selection does not utilize some of the evidence about human processing during reading that suggests feature extraction occurs in parallel with the development of an understanding of the text. Feature extraction in human reading is a two-step process that can be framed as hypothesis generation and testing. The understanding process includes syntactic as well as semantic components.

This paper presents a set of algorithms for text recognition that model the essence of human reading with two feature extraction stages and an understanding phase that uses information about the syntactic context between words. An objective is to discover how different feature sets affect the performance of syntax.

Statistical experiments show that a simple representation for syntax reduces the number of words in a large lexicon that can match an input word by about 20 percent. Also, the error rate is reduced as the power of the feature detectors is increased.

1. INTRODUCTION

Text recognition algorithms are often designed to work in a bottom-up fashion. A fixed set of features are extracted from each character and recognition is performed by classifying these features. Sometimes, a sequence of decisions is post-processed and the word in a dictionary that best matches them is chosen and output. This effectively uses contextual information within words to improve the recognition rate. However, it ignores valuable information about context *between* words of text.

The human reader differs from a text recognition algorithm in many ways. People develop an understanding of a text while reading it. The recognition of symbols in the text is of secondary importance. In fact, the human reader tries to minimize the effort expended in extracting features by developing a strategy for feature analysis in which, basically, a small number of features are extracted that can be effectively used to understand the text [1]. Also, the features are extracted selectively from words rather than from every character as is done by most text recognition algorithms.

The organization of the human reading process has been investigated by many psychologists. Two stages of feature analysis have been identified in human reading [1]. A *bottom-up* phase extracts a simple set of features that are used to develop hypotheses about the identity of a

word. When a running text is read, this analysis also provides information that guides eye movements. A *top-down* phase of feature extraction uses the information supplied by the bottom-up stage as well as the understanding that is being developed as the text is being read to test for some additional features. The results of these tests determine the words that are perceived. The two feature analysis steps are controlled by a third level of cognitive processing that integrates their results and develops an understanding of the text, provides expectations about what will be seen next, and helps to direct the eye movements of the reader. The third level utilizes many sources of knowledge such as syntactic and semantic [2] information and is influenced by many factors such as the mood of the reader [3].

There are many aspects of human reading that can be applied to the development of text recognition algorithms even though the goals of the two processes are different. One important point is that feature extraction in human reading is performed on words. The closest analog in computer vision are whole-word recognition algorithms that they have frequently been applied to cursive script recognition (e.g., [4]). An important characteristic of such approaches is that they use knowledge from a vocabulary of words during recognition rather than postponing its use until after critical decisions have been made. This is closer to the way people do feature analysis than a method that individually recognizes characters. Another aspect of human reading that applies to text recognition algorithms is the influence of high-level knowledge on feature analysis. The use of information about syntax and semantics to influence feature extraction is one way that text recognition algorithms can be made more tolerant to image noise, differences in font, and so on. Some efforts have been made in this area (e.g., [5, 6]), but there are still many avenues still open to exploration.

The remainder of this paper is concerned with the use of high-level syntactic information to influence feature extraction. A text recognition algorithm is presented that is based on human performance. The algorithm contains two stages of feature analysis that operate on word images and a high-level analysis step. A representation for the syntax of an input language is proposed and its ability to influence the feature analysis stages is explored with statistical experiments. It is shown that syntax can significantly improve the performance of the text recognition algorithm and that the extent of the improvement depends on the feature set that is used.

2. TEXT RECOGNITION ALGORITHM

The recognition algorithm contains the three stages and interactions between them shown in Figure 1. The input is a sequence of word images w_i, $i = 1,2, \cdots$. The hypothesis generation stage corresponds to the bottom-up stage of visual processing in human reading. A gross visual description of w_i is computed and a set of words or hypotheses N_i (called the *neighborhood* of w_i) is looked up in a dictionary that is assumed to contain the input word. Hypothesis generation succeeds if the input word is contained in N_i.

The hypothesis testing phase of the algorithm corresponds to the top-down stage of visual processing in human reading, i.e., goal-directed feature testing. Hypothesis testing uses the contents of a neighborhood to determine a sequence of feature tests that could be executed to recognize w_i. This sequence can begin at several locations in w_i and consists of a variable number of tests. The output of hypothesis testing is either a unique recognition of w_i or a set of hypotheses that contains the word in w_i.

The global contextual analysis phase of the algorithm corresponds to the cognitive processes of human reading. Information from other words that have been recognized in the text and some other global contextual information is used to constrain the words that can be in N_i. The output is a neighborhood N_i^* of reduced size.

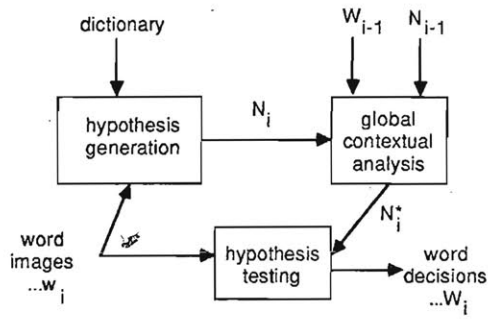


Figure 1. Organization of the algorithms.

2.1. Hypothesis generation

The hypothesis generation stage of the algorithm computes a feature description of an input word. This feature description is used to access a dictionary and return a set of words or neighborhood that have the same feature description as w_i.

The feature description should be simple and require only rudimentary image processing techniques. This is to increase the accuracy of the approach and make it tolerant to variations in image noise, font style, size, and so on.

In this paper, lower case machine-printed text was of primary concern because it offers more informative and easily computed features than does upper case. This is demonstrated in psychological experiments where it takes people about 10 percent longer to read text printed completely in upper case than text printed in mixed upper and lower case [7]. One of the feature descriptions for lower case text that was used included three types of vertical bars that occur in many characters. These are the ascender (e.g., as in the 'b'), the descender (e.g., as in the 'p'), and the short vertical bar (e.g., as in the 'r'). Various other features were also used (e.g., the dot over a short vertical part that occurs in an 'i').

An example of calculating the neighborhood for the image of the word *may* is to first determine that it contains four short vertical bars (the character 'y' contains no feature). Other words that have the same feature description are *me*, *now*, and *over*. These four words are the neighborhood of the input image.

2.2. Hypothesis testing

The hypothesis testing routine implements a goal-directed feature testing of an input image. A complete description of the technique is given elsewhere [8]. A short description is given here. The words in the neighborhood of an input word determine the features that could occur at specific locations. This knowledge determines a discrimination test that can be executed on the image at that location. Because some features can be common to several words, the result of a discrimination test is either a unique recognition or a reduction in the number of words that

could match the image. An example of this is given by the neighborhood *me*, *now*,*may*, and *over*. The area between the first two short vertical bars can be either "closed at the top only" or "closed both at the top and bottom". In the first case the words that could match are reduced to *me*, *now*, and *may*. In the second case the input is recognized as *over*.

When the result of a discrimination test is not a complete recognition of the word, the features that could occur in other locations are further constrained. This determines other discrimination tests. In the above example, if the area between the first and second short vertical bars is closed at the top, the area between the second and third short vertical bars can be either closed at the top (for *me* and *may*) or it can be a small empty space (for *now*). If *over* were not eliminated by the first test, this area could contain another feature: a large non-empty space (containing a *v*).

The process of constraint generation followed by discrimination tests is repeatedly applied until the word is recognized. The process can begin at several locations within a word and can utilize a variable number of discrimination tests depending on that starting location. The series of tests and their locations is computable a-priori and can be used in several alternative recognition strategies. For example, it might be desired to execute the fewest number of discrimination tests. A starting location could be chosen to accomplish this goal. This might result in more complicated discriminations but would require less computation than another sequence of tests. Another strategy might use simpler discriminations at the expense of more computation. This could be more reliable than the other strategy.

2.3. Global contextual analysis

The global contextual analysis phase of the technique reduces the number of hypotheses that could match a word. Some characteristics of the previous word and some knowledge about the text being recognized are used to do this. Several forms of global contextual knowledge have been used before [5]. These rely on a table of allowable transitions to constrain the words that can follow other words. Words are removed from a neighborhood that are not legal according to the table. Word-to-word and feature description-to-word transitions were considered. The word-to-word transitions provided the best performance. However, an impractically large sample of text was needed to compile the transition tables. The feature description-to-word transitions had the same drawback.

Global contextual analysis uses knowledge beyond the word level to improve recognition performance. This is a routine part of human reading that has not been frequently imitated in text recognition algorithms. The work discussed in this paper is a limited approximation of the many high level processes that a person executes when reading and that are encompassed by this stage of the algorithm. This work indicates the usefulness of such information and provides impetus for future investigations.

3. SYNTACTIC CONTEXT

The syntactic context of a word of text refers to the syntactic classes of nearby words, where syntactic classes are categories such *noun*, *verb*, *adverb*, and so on. A sentence is syntactically correct if the sequence of classes of its words can be generated by a grammar for the underlying language. It should be noted that a sentence can be syntactically correct but still be nonsensical. An example is:

Colorless green ideas sleep furiously.

This shows that syntax is only one component needed to understand a text. However, it is a very important knowledge source since it can detect obvious inconsistencies such as:

The dog at the cat.

The implication of this for feature extraction in text recognition is that syntax can indicate inconsistencies and focus attention on areas in the text that can be analyzed to resolve them.

For this to work at all, there must be some way to assign syntactic classes to words. In natural language processing systems this is sometimes done by parsing an input sentence according to a grammar for the language [9]. Many approaches can be used including transformational, systemic, case, or other types of grammars and chart, augmented transition network, situation-action, or other kinds of parsers.

Even though most of these grammars and parsing techniques are quite powerful, most of them require that a grammar be manually specified and that a lexicon be supplied that lists the syntactic classes of the words that appear in the text. These problems might be solved by grammatical inference or automatic tagging techniques [10]. However, it is unclear how well such methods could be applied to a text recognition algorithm where the words of text can only be approximately recognized.

3.1. Representation for Syntax

To overcome the need to manually specify a complete grammar for the English language, a more modest representation for syntax is adopted. The sequence of syntactic classes that occur in a running text is approximated by transitions between words and the classes that can follow them. This is a crude substitute for a grammar. However, it has the advantage that it can be easily compiled by a single pass through a large, pre-classified sample of text and does not require a complex parsing mechanism.

3.2. Usage of the Representation

The transitions are used in the global contextual analysis portion of the text recognition algorithm. It is assumed that the $i-1^{st}$ word in the text has been recognized when the ith word is processed. It is also assumed that a dictionary is given that includes all the words that could occur in the text and their syntactic classes. Note that each instance of a word in a running text has a distinct class but that multiple instances of a word can have different classes. All the classes that could appear with a word are represented in the dictionary.

A word is removed from the neighborhood of the ith word if all its classes cannot follow the $i-1^{st}$ word. The smaller neighborhood that is the result of this process is then passed to the hypothesis testing algorithm.

A simple example of how this works is illustrated by the phrase *a toy*. If the neighborhood for *toy* is {*the, toy*}, *the* would be removed if it were known that an article could not follow *a* in the text.

4. STATISTICAL MEASUREMENT OF EFFECTIVENESS

The effectiveness of the representation for syntactic context is measured by how much it reduces the neighborhood size of a word. This can be determined by computing the following statistic before and after the syntactic constraint is employed:

$$ANS_t = \frac{1}{N_t} \sum_{i=1}^{N_t} ns(tw_i)$$

where N_t is the number of words in the text and $ns(tw_i)$ is the size of the neighborhood for the ith word of text. This is calculated:

$$ns(tw_i) = \sum_{j=1}^{N_d} eqshape(tw_i, dw_j),$$

where N_d is the number of words in a dictionary (list of unique words), dw_j is the jth dictionary word, and

$$eqshape(\alpha,\beta) = \begin{cases} 1 & \text{if } feat\_desc(\alpha) = feat\_desc(\beta) \\ 0 & otherwise, \end{cases}$$

where *feat_desc* is the feature description computed by the hypothesis generation algorithm. Examples of features descriptions are the number of characters in a word, the sequence of ascenders and descenders it contains, and so on.

ANS_t provides a measurement of the average number of dictionary words that are input to the hypothesis testing routine. The performance of the text recognition algorithm will improve if this value is decreased because the hypothesis testing routine will have fewer alternatives to decide among. Therefore, the extent of the reduction in ANS_t provided by the syntactic constraints is a measurement of their effectiveness.

4.1. Text Database

A series of experiments were conducted in which ANS_t was computed from subsets of the Brown Corpus [11]. This text was chosen because it is large (over 1,000,000 words of running text) and every word is tagged with its syntactic class. The corpus is divided into 15 subject categories or genres. There are 500 individual samples of running text in the corpus and each one contains approximately 2000 words. The number of samples in each genre differs depending on the amount published in that area at the time the corpus was compiled. The genres and the number of samples in each one are listed in Table 1. A sample was randomly chosen from each genre for experiments that will be discussed later. The number of this sample is also shown in Table 1.

genre	category	no. samples	chosen sample
A.	Press Reportage	44	6
B	Press Editorial	27	23
C	Press Reviews	17	12
D	Religion	17	5
E	Skills	36	2
F	Popular Lore	48	33
G	Belles Lettres	75	20
H	Miscellaneous	29	14
J	Learned	80	9
K	General Fiction	29	26
L	Mystery and Detective Fiction	24	21
M	Science Fiction	6	3
N	Adventure and Western Fiction	29	8
P	Romance and Love Story	29	14
R	Humor	9	9

Table 1. The genres and the number of samples in each genre of the Brown Corpus. A randomly chosen sample in each genre is also indicated.

There are 81 different tags that fall into six major categories:

(1) *parts of speech:* nouns, verbs, adjectives, and so on.

(2) *function words:* determiners, prepositions, conjunctions, and so on.

(3) *important individual words: not*, existential *there*, infinitival *to*, forms of *do*, *be*, and *have*.

(4) *punctuation marks of syntactic importance:* ".", "(", ")", "--", ",", ":".

(5) *inflectional morphemes:* noun plurals, and possessives, verb past, present and past participle, and so on.

(6) *foreign or cited words.*

A tag sometimes has a '*' or a '$' affixed to it to indicate the word is a negation or possessive form, respectively. These distinctions were retained since they usually imply a change in the appearance of the word. This yielded the set of tags shown in the Appendix.

4.2. Feature Descriptions

Three features descriptions were used to discover the effect of differing the feature set on the reduction in ANS_t. The feature descriptions were specialized for lower case for the reasons discussed earlier. However, this does not limit the applicability of the results to lower case text because a feature description for mixed upper and lower case has been shown to produce as good performance in hypothesis generation as feature descriptions for lower case only [12].

The first feature description is the number of characters in a word. This is sometimes easy to compute and is the same whether the word contains all lower case letters or a mixture of upper and lower case. This description was used since it is easy to compute if the characters in a word do not touch and can be reliably estimated if the incidence of touching characters is low.

The second feature description is specialized for lower case characters. It includes different heights of vertical bars, dots, and empty spaces. These features were chosen because they can be reliably computed from images of text even if the characters touch one another [13]. The complete listing of this feature set is given below:

1. A significant area at the beginning or end of a word that does not contain a vertical bar (e.g., the space to the right of the vertical bar in a "c");
2. A short vertical bar (e.g., the leg of an "r");
3. A long high vertical bar that extends above the main bar of the word (e.g., the ascender portion of a "b");
4. A long low vertical bar that extends below the main body of the word (e.g., the descender in a "p");
5. Dots over short vertical bars (occurs in an "i");
6. Dots over long vertical bars (occurs in a "j").

The third feature description includes the features in the second plus a feature for holes. The holes were used since they increase the discriminatory power of the feature description. However, it may be difficult to reliable detect holes in practice. This feature is described as:

7. The hole that occurs in "a", "b", "d", "e", "g", "o", "p", or "q".

When the second and third feature descriptions are applied to a word they yield a symbolic representation that is the sequence of occurrence of the features in the image. Thus, "may" has a symbolic representation 22221 with the second feature description and 222721 with the third feature description.

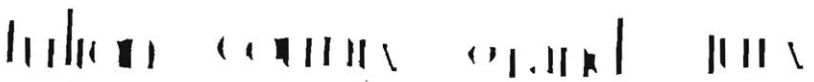
4.3. Example

An example of the operation of the algorithm is shown in Figure 2. Figure 2(a) shows an image of the phrase *fulton county grand jury* and (b) shows the vertical bars extracted from the image. The original neighborhoods are shown in (c). These were computed by finding the words in a dictionary that have the same features as those shown in (b). The reduced neighborhoods obtained after the application of the syntactic constraint are shown in (d). The syntactic classes that can follow the words in (a) are shown in (e).

The word greater is removed from the neighborhood for county because its class (JJR) is not among the classes that can follow fulton (JJ,NN,NN$,NNS). The classes associated with the other three words are all in this list. The word urged is removed from the neighborhood for grand for similar reasons.

In this example, ANS_t = (1+4+3+1)/4 = 2.25 before the neighborhoods were reduced and ANS_t = (1+3+2+1)/4 = 1.75 after the reduction. Thus, ANS_t was reduced by 22 percent.

(a) fulton county grand jury

(b) 

(c)	fulton NP	greater JJR money NN county NN matters NNS	grand JJ good JJ urged VBD	jury NN
(d)	fulton	money county matters	grand good	jury
(e)	JJ NN NN$ NNS	, . HVZ JJ MD NN VBG	NN NNS	. DOD HVD RB RBR VBD

Figure 2. An example of using syntactic context. An input image (a) and the vertical bars in it (b) are shown. Also, the original (c) and reduced (d) neighborhoods are given as well as the grammatical classes for each word (e).

4.4. Experiments

The change in ANS_t with the addition of syntactic context was measured for each of the feature descriptions from a randomly chosen sample of text in each of the 15 genres. These are the samples indicated in Table 1. The dictionary was the same for each run and included all the words in the corpus. This was consistent with the assumption expressed earlier that all the words that could appear are known. However, the syntactic tags associated with each word were not compiled by inspection of the sample from which the statistics were calculated. The tags were determined from the entire corpus minus the selected sample. The same procedure was used to calculate the grammatical tags that follow the words. This methodology was used to avoid testing on the training data.

Because the dictionary and transition tables were assembled in this way, it is possible for errors to occur. This happens when a correct word is removed from its neighborhood. If this results in a neighborhood of size zero, then the original neighborhood size, before it was reduced, is used in the calculation of ANS_t. If the size of a neighborhood with an error is greater than zero, this size is still used since this is an accurate reflection of the behavior of the algorithm in practice.

The results of the experiments are given in Table 2. The results show that the first feature description, that calculated the number of characters in a word, had the worst overall performance with an average ANS_t of 2932 before syntactic context was used and 2384 after. This is not surprising because so few features were employed. However, it is interesting that syntax was still able to reduce ANS_t by about 19 percent. The second feature description had much better overall performance with an average ANS_t of 39 before and 31 after syntax was used. The much improved performance is because of the additional features. It should also be noted that the average error rate dropped by 39 percent from 5.7 percent with the first feature description to 3.5 percent with the second. This is a reflection of the increased discriminatory power under the second feature description, i.e., an error is more likely to be detected because the reduced neighborhood output by global contextual analysis is more likely to contain zero words. The third feature description had similar performance. The average ANS_t was 20 percent and the average error rate was reduced by 51 percent.

5. SUMMARY, CONCLUSIONS, AND FUTURE WORK

The interaction between feature extraction and language syntax in a text recognition algorithm based on human performance was explored. A representation for the syntactic context between words of text was proposed. It was used to reduce the number of words in a large dictionary the could match the image of a word of text. The performance of reduction process was tested with statistical experiments on a large sample of text. Several feature descriptions were used to perform the matching.

The experiments showed that a simple representation for syntax was able to reduce the average number of matches by about 20 percent. The more precise feature descriptions (those that used more features) had lower error rates. However, the descriptions are more difficult to compute as their precision increases. These results indicate the existence of a tradeoff between the choice of a feature description and the performance of the syntactic constraints. Syntax will provide more assistance as the precision of the description is increased. But increasing the precision requires more powerful feature extraction techniques.

Future work on the approach discussed in this paper will concentrate in two areas. Application of the method to digital images of text will be pursued and a probabilistic representation for syntax will be developed to better accommodate imprecise information like that provided by feature extraction routines. The probabilistic method will be applied to entire sentences to find the best match between the images of words and entries in the dictionary.

genre	feature description 1				feature description 2			
	ANS_t before	ANS_t after	percent improve.	percent errors	ANS_t before	ANS_t after	percent improve.	percent errors
A	2947	2360	20	6	38	29	24	4
B	2945	2406	18	5	39	32	18	3
C	3128	2524	19	6	37	29	22	4
D	2999	2431	19	7	37	29	22	4
E	2856	2374	17	6	39	31	21	5
F	2930	2390	18	4	38	30	21	2
G	2899	2377	18	6	41	32	22	3
H	3379	2734	19	5	38	30	21	3
J	3064	2501	18	7	30	23	21	4
K	2560	2134	17	5	42	34	19	3
L	2867	2336	19	5	40	32	20	3
M	2814	2356	16	4	41	34	18	3
N	2973	2353	21	7	40	32	21	4
P	2745	2191	20	4	42	34	19	3
R	2875	2293	20	8	38	31	20	5
avg.	2932	2384	19	5.7	39	31	21	3.5

genre	feature description 3			
	ANS_t before	ANS_t after	percent improve.	percent errors
A	4.9	3.7	24	2
B	4.8	3.9	19	1
C	4.7	3.8	21	2
D	4.7	3.8	19	1
E	5.1	4.2	18	3
F	5.2	4.1	21	1
G	4.9	3.8	22	2
H	4.6	3.6	23	2
J	4.6	3.7	20	2
K	5.1	4.2	17	1
L	4.9	4.0	19	1
M	5.1	4.1	19	2
N	4.5	3.6	20	2
P	4.9	4.1	16	2
R	5.2	4.3	17	2
avg.	4.9	3.9	20	1.7

Table 2. The effects of the feature sets.

ACKNOWLEDGMENTS

This work was supported by the Office of Advanced Technology of the United States Postal Service. The author would like to thank Dr. Tim Barnum and Gary Herring of USPS and Dr. John Tan of Arthur D. Little, Inc. for their assistance.

REFERENCES

[1] J. Hochberg, "Components of literacy: Speculations and exploratory research," in *Basic Studies on Reading*, H. Levin and J. P. Williams (editor), Basic Books, Inc., New York, 1970, 74-89.

[2] K. Rayner, M. Carlson and L. Frazier, "The interaction of syntax and semantics during sentence processing: Eye movements in the analysis of semantically biased sentences," *Journal of Verbal Learning and Verbal Behavior 22*, 3 (June, 1983), 358-374.

[3] S. A. Small, "The effect of mood on word recognition," *Bulletin of the Psychonomic Society 23*, 6 (1985), 453-455.

[4] R. F. H. Farag, "Word-level recognition of cursive script," *IEEE Transactions on Computers C-28*, 2 (February, 1979), 172-175.

[5] J. J. Hull, "Inter-word constraints in visual word recognition," *Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence*, Montreal, Canada, May 21-23, 1986, 134-138.

[6] H. S. Baird and K. Thompson, "Reading chess," *Proceedings of the IEEE Computer Society Workshop on Computer Vision*, Miami Beach, Florida, November 30-December 2, 1987, 277-282.

[7] D. F. Fisher, "Reading and visual search," *Memory and Cognition 3*, 2 (1975), 188-196.

[8] J. J. Hull, "Hypothesis testing in a computational theory of visual word recognition," *Sixth National Conference on Artificial Intelligence* , Seattle, Washington, July 13-17, 1987, 718-722.

[9] T. Winograd, *Language as a cognitive process*, Addison-Wesley Publishing Company, Reading, Massachusetts, 1983.

[10] L. L. Cherry, "PARTS - A system for assigning word classes to English text," Computing Science Technical Report No. 81, Bell Laboratories, June 12, 1978.

[11] H. Kucera and W. N. Francis, *Computational analysis of present-day American English*, Brown University Press, Providence, Rhode Island, 1967.

[12] J. J. Hull, "A computational theory of visual word recognition," Technical Report, State University of New York at Buffalo, Department of Computer Science, February, 1988.

[13] J. J. Hull, "Hypothesis generation in a computational model for visual word recognition," *IEEE Expert 1*, 3 (Fall, 1986), 63-70.

APPENDIX: SYNTACTIC TAGS USED IN THE EXPERIMENTS

(	left paren	CC	coor. conj.	IN	preposition	PPLS	pl. ref./int. pers. pro.
)	right paren	CD	card. num.	JJ	adj	PPO	objective pers. pro.
*	*not, n't*	CD$	poss. card. num.	JJ$	poss. adj	PPS	3rd nom. pro.
,	comma	CS	sub. conj.	JJR	comp. adj	PPSS	other nom. pers. pro.
--	dash	DO	*do*	JJS	sem. super. adj	QL	qual
.	sent. close	DO*	*don't*	JJT	morph. super. adj	QLP	post-qual
:	colon	DOD	*did*	MD	modal auxiliary	RB	adverb
ABL	pre-qualifer	DOD*	*didn't*	MD*	negative auxiliary	RB$	poss. adverb
ABN	pre-quant	DOZ	*does*	NN	s. or mass n	RBR	comp. adverb
ABX	pre-quant	DOZ*	*doesn't*	NN$	poss. or mass n	RBT	super. adverb
AP	post-det	DT	det	NNS	pl. n	RN	noml. adverb
AP$	poss. AP	DT$	poss. det	NNS$	poss. pl. n	RP	adverb/particle
AT	article	DTI	det/quant	NP	prop. n	TO	infinitive marker *to*
BE	*be*	DTS	pl. det	NP$	poss. prop. n	UH	interjection, excl.
BED	*were*	DTX	det/dbl conj.	NPS	pl. prop. n	VB	verb, base form
BED*	*weren't*	EX	exist. *there*	NPS$	poss. pl. prop. n	VBD	verb, pst tense
BEDZ	*was*	FW	foreign word	NR	adverbial n	VBG	verb, pres. part./ger.
BEDZ*	*wasn't*	HV	*have*	NR$	poss. adverbial n	VBN	verb, pst part.
BEG	*being*	HV*	*haven't*	NRS	pl. adverbial n	VBZ	verb, 3rd. pres.
BEM	*am*	HVD	*had* pst tense	OD	ordinal number	WDT	*wh*- det
BEM*		HVD*	*hadn't*	PN	noml. pro.	WP$	poss. *wh*- pro.
BEN	*been*	HVG	*having*	PN$	poss. noml. pro.	WPO	objective *wh*- pro.
BER	*are, art*	HVN	*had* pst part.	PP$	poss. pers. pro.	WPS	nom. *wh*- pro.
BER*	*aren't*	HVZ	*has*	PP$$	2nd noml. poss. pro.	WQL	*wh*- qual
BEZ	*is*	HVZ*	*hasn't*	PPL	ref./int. pers. pro.	WRB	*wh*- adverb
BEZ*	*isn't*						

important abbreviations: adj = adjective, card. = cardinal, comp. = comparative, corr. = coordinating, det. = determiner, int. = intensive, n = noun, nom. = nominative, noml. = nominal, part. = participle, pers. = personal, pl. = plural, poss. = possessive, pro. = pronoun, prop. = proper, pst = past, qual. = qualifier, quant = quantifier, ref. = reflexive, s. = singular, sub. = subordinate, super. = superlative,

FROM PIXELS TO FEATURES

Proceedings of a Workshop held at
Bonas, France, 22-27 August, 1988

edited by

J.C. SIMON

Université Pierre et Marie Curie - Paris VI
France

1989

NORTH-HOLLAND
AMSTERDAM • NEW YORK • OXFORD • TOKYO