Sentence Boundary Detection:
A Comparison of Paradigms for Improving MT Quality

Daniel J. Walker, David E. Clements, Maki Darwin and Jan W. Amtrup
Mendez, Inc.
A Division of Lernout & Hauspie, N.V.
5095 Murphy Canyon Road, Suite 300
San Diego, CA 92123, USA
{dwalker,dclements,mdarwin,jamtrup}@lhsl.com

Abstract

The reliable detection of sentence boundaries in running text is one of the first important steps in preparing an input document for translation. Although this is often neglected, it is necessary to obtain a translation with a high degree of quality. In this paper, we present a comparison of different paradigms for the detection of sentence boundaries in written text. We compare three different approaches: Directly encoding the knowledge in a program, a rule-based system relying on regular expressions to describe boundaries, and a statistical maximum-entropy learning algorithm to obtain knowledge about boundaries. Using the statistical system, we obtain a recall of 98.14%, classifying boundaries of six types, and using a training corpus of under 10,000 sentences.

Keywords

Segmentation, MT for the Web, Maximum-Entropy models, MT Methodologies, Evaluation

Introduction

The division of running input text into sentences that can be translated in isolation is one of the first important steps for any natural language processing system. In particular, any machine translation system needs reliable information about sentences that present a coherent syntactic portion of text to be analyzed.

Consider the following examples of text:

(1) *Is K.H. Smith here?*

(2) *I bought the apples, pears, lemons, etc. Did you eat them?*

Two types of problems occur frequently: splitting a single sentence into fragments and falsely conjoining two separate sentences into a single segment. In both cases, our current MT system (L&H Power Translator Pro 7 English-to-French) attempts to parse the ungrammatical input, resulting in poor translation quality.

The first sentence translates as

(3) *Est K.H. Smith ici?*

The sentence is split into two on the unknown abbreviation, *K.H.* The translator parses <Is K.H.> as one unit, and <Smith here?> as another. If the input to Power Translator is a similar sentence without an abbreviation, the output is correct French:

(4) *Est-ce que Kevin Smith est ici?*

Example (2) above shows the other problem, when two separate sentences are incorrectly conjoined into a single segment. Common abbreviations are kept in a table that the MT engine consults when attempting to disambiguate sentence boundaries. However, because the period is included within the tokenized abbreviation, the parser fails to recognize a sentence boundary when that abbreviation occurs at the end of a sentence.

Power Translator joins the two segments and attempts to translate them together:

(5) *Est-ce que j'ai acheté les pommes, poires, citrons, etc. est-ce que vous les avez mangés?*

Both sentences are translated as a single question, because of the question mark at the end of the second. If one removes the ambiguous sentence boundary, *etc.*, the MT engine produces a correct parse of both segments:

(6) *J'ai acheté les pommes, poires, citrons, etc.! Est-ce que vous les avez mangés?*

These two simple examples shall suffice to demonstrate the importance of correct sentence boundary detection. However, there is only a relatively small body of published research about this topic, partly because it is considered a side issue for many research systems that assume the input already is divided into sentences, partly because it is often considered a mundane task that can be reasonably carried out with the application of a few regular expressions.

Our experience is with translation of texts of a high variety of sources and sorts, including HTML pages, e-mail, inline chat clients and Microsoft Word documents. This experience suggests that the quality of sentence boundary disambiguation, and the evaluation of different approaches, deserves more attention. In this paper, we present three different paradigms for this task:

- The direct incorporation of knowledge about boundaries into a translation system, without reference to any higher level of description,
- The representation of sentences using regular expressions, which divide a text, and

- The application of machine learning techniques (in particular, a maximum-entropy approach) to the task in question.

In comparing the systems, we are mostly interested in high recall. Each boundary occurring in a text should be recognized. High precision (assigning only true boundaries) is also important, but we do not regard it as essential. The experience with our MT systems suggests that a system produces translations of higher quality when facing a sentence that has been divided into fragments as compared to translations of segments that contain more than one sentence.

Additionally, we regard development time for a model as an important factor. Reducing development time reduces the cost of developing and improving commercial translation systems. We are in favor of models that only need a relatively short time to implement and maintain.

The Direct Model

The MT system we are currently using (the Barcelona engine as part of the Power Translator product) uses a hard-coded routine to detect sentence boundaries. The algorithm is inspired by regular expression techniques; however, no higher level modeling of the relevant knowledge has been employed. All processing is implemented directly. It uses an abbreviation lexicon to improve the accuracy of its operation. The development time for the system was between one and two person-months, resulting in monolithic code with some special handling for idiosyncratic marking in individual languages. Though efficient and reasonably accurate (see below for results), this kind of implementation poses a maintenance problem: Each change in behavior means modifying the engine code, a cost-intensive and error-prone process. The extension of the system to another language in practice often means revisiting all of the segmentation code, yielding the same investment as for the initial language in the worst case. Moreover, for each change, two persons have to be involved, a linguist to describe the realities, and an engineer to encode them into the program.

Rule-Based Disambiguation

In order to alleviate the problems mentioned for the direct implementation, the decision was made to separate the description of sentence boundaries and the processing needed for their recognition. For this prototype, a grammar describing sentence boundary detection has been written. We use regular expressions to represent the properties of sentences. For instance, the rule

(7) *Sentence -> All Word PERIOD PUNCT PERIOD;*

describes a sentence consisting of a number of words, followed by a period, an unspecified punctuation character and another period. This rule would then be able to capture ellipsis and other combinations like .”. Regular expressions over characters are used to describe particular entities within a sentence. For instance,

(8) *'[\-_A-za-z0-9.]+\.{dom}' WEBADDRESS;*

is one of the ways we use to describe Internet addresses.

The processing needed to match a complete grammar of regular expressions is basically of the lex/yacc style. Although theoretically capable of recognizing context-free languages, in practice the descriptions are only regular.

Compared to the direct model, this kind of representation has the obvious advantage that linguistic knowledge is encoded in a declarative way, making it relatively easy to change the behavior of the model, and to add further languages. Adding a language only requires the development of a new grammar, still a considerable effort, but much easier than reviewing and partially rewriting the code for the direct model. Plus, this kind of model increases the efficiency of linguists by allowing them to write the grammar, independent of any coding an engineer would implement in the MT engine under the direct model. The initial investment for the development of this system was slightly higher than for the aforementioned method, approximately 3 person-months, adding new languages should be considerably faster.

The Maximum-Entropy Model

The third method we employ views the problem of identifying sentence boundaries as a statistical classification problem. Potential boundaries may be classified as actual boundaries according to features of the context in which they appear. Reynar and Ratnaparkhi (1997) show that Maximum Entropy Models may be employed to classify sentence boundaries with high accuracy.

The fundamental principal behind Maximum Entropy Modeling is that the likelihood of a certain class, in our case sentence boundaries, appearing in a given context can be estimated by the probability distribution with maximal entropy subject to certain constraints. The Maximum Entropy Model considers only specific evidence of sentence boundaries in the text. This evidence represents prior linguistic knowledge about contextual features of text that may indicate a sentence boundary and are determined by the experimenter. Since the model only considers the distribution of these explicitly identified features, all other features of the text are assigned a uniform distribution. Thus, the model is "maximally" uncertain about features of text for which it does not have prior knowledge.

We essentially reproduce the model described by Reynar and Ratnaparkhi (1997). The model evaluates the context of each candidate sentence boundary via several linear functions each of which indicates whether or not a given context has a particular feature. The model is constrained in such a way that the expectation of each feature in any context is the same as the observed expectation of that feature in the training data.

(9) $$E_p f(c,l) = \sum_{c,l} p(c,l) f(c,l)$$

$$\textbf{(10)} \quad E_p f_i(c,l) = E_{\tilde{p}} f_i(c,l)$$

This constraint is implemented by weighting the value of each feature. The model has the form:

$$\textbf{(11)} \quad p(c,l) = \frac{1}{Z} \sum_{f_i} a_i^{f_i(c,l)}$$

where c is the context in question, l ∈ {boundary, non-boundary}, and a is the weight of each feature, f.

The weights are the unknown parameters of the model and must be discovered during training. We employ the Generalized Iterative Scaling (GIS) algorithm by Darroch and Ratcliff (1972), which finds the distribution of the form above with maximal entropy under our constraints. See Ratnaparkhi (1997) for a complete description of the algorithm.

Calculating the expectation of the features is computationally expensive since each feature must be evaluated several times. The task must be performed for each iteration and hundreds of iterations may be required before the weights have satisfactorily converged. However, since the values of features are constant for each context, we took the opportunity to store these values in a cache and simply retrieve them during successive iterations. We found that this decreased training time several fold. In general, we found that training time is more strongly dependent on the number of features and the granularity of weight convergence than on the size of the corpus.

We identified 15 simple features such as whether the candidate contains a period or question mark, whether the candidate is on a list of abbreviations extracted from our corpora, capitalization, etc. We considered only the words immediately preceding and following the candidate. The experience of Reynar and Ratnaparkhi (1997) suggests that considering a broader context does not improve the model's performance.

The accuracy of the model in identifying sentence boundaries relies on the quality of the prior linguistic knowledge embedded in the system. We selected features though experimentation. When the model failed to classify sentence boundaries, we attempted to identify contextual evidence that the model may need to consider. Then we re-trained the model with the additional feature and gauged the affect.

The features that we considered were typically very simple to identify without requiring any sort of grammar or elaborate regular expression. Though certain features initially appear to conflict, their frequencies in the corpora constrain the model's expectations of them. For example, consider these two features:

- Words containing sentence boundaries will have no characters following the boundary mark.
- A period followed by a quotation mark indicates a sentence boundary.

We found that if the model hasn't been adequately trained or if the training data didn't contain substantial evidence for feature 2, it would rarely classify periods followed by quotation marks as boundaries. However, with more training data or by choosing a smaller granularity of weight convergence, the model would more consistently make correct classifications.

The development of this method was relatively quick: We developed an annotation tool in approximately two person-weeks. The training algorithm, together with the features that we used, took another person-week to develop. The annotation of the training and test data also took approximately one person-week.

Related work

Palmer and Hearst (1997) describe a system using the syntactic context of a potential sentence boundary to classify the boundary. The boundary and the parts of speech of a number of context words (six words on each side) are fed to a neural network that determines the function of the boundary character. They report an accuracy between 98.5% and 98.9% on Wall Street Journal (WSJ) data, depending on the size of the training and test data. Their system requires the data to be tagged with parts of speech, which poses a circularity problem, as POS taggers usually require prior segmentation. Mikheev (2000) solves this problem by performing segmentation during POS tagging, further enhanced by a method for the disambiguation of proper nouns.

Grefenstette and Tapanainen (1994) use regular expressions augmented with linguistic knowledge about abbreviations (their formation and a lexicon of frequent abbreviations) to detect boundaries; they achieve a sentence recognition rate of 99.07% for sentences that end with a period.

The maximum-entropy model we are using for the comparison in this paper is due to Reynar and Ratnaparkhi (1997). Using context information geared towards financial newspaper text and a training corpus of about 40,000 sentences, they achieve an accuracy of 98.8% on WSJ data (the corresponding, more general model delivers an accuracy of 98.0%).

Experiments and Results

We collected a small corpus of 96 documents from various web sites. These documents contain a total of 10365 sentences, which were manually annotated with sentence boundaries. The characters used as boundary indicators are period, question mark, exclamation mark, colon, semicolon, and the closing parenthesis. We held out 11 randomly selected documents, containing 861 boundaries and 215 non-boundaries for testing. The remainder with 9504 boundaries and 3048 non-boundaries was used as training material. Markup in the training data was removed, but it was not further preprocessed or cleaned in any way, e.g. to resolve spelling errors or formatting issues. Table 1 shows the results of analyzing the source documents. We report results both for the training and the test set. For the maximum-entropy system (Entr) we can thus estimate the degree of degradation when processing unseen text. For the other systems, the results should be comparable, as they were developed before the selection of the corpus. The table shows the number of correctly annotated boundaries (Ok), the number of insertions (Ins) and deletions (Del), as well as precision (Prec), recall (Rec) and the F-measure (F) with $\alpha = 0.5$. The systems are: Dir) the direct method, Rul) the rule-based system, and Entr) the maximum-entropy model. Since both the direct and rule-based system were designed to insert sentence boundaries in running text not only at the special sentence boundary markers we chose, but also elsewhere (for instance, at line-breaking characters), we also give results that reflect only the boundaries that occur at the markers. These results are given as systems Dir*) and Rul*), respectively.[1] The system marked ***) represents a baseline performance, assuming that every potential sentence boundary character indeed marks a sentence boundary.

		Ok	Ins	Del	Prec	Rec	F
Training	***	9504	3048	0	75.72	100.00	86.18
	Dir	6504	239	956	96.46	87.18	91.51
	Dir*	6504	134	956	97.98	87.18	92.27
	Rul	8982	1806	522	83.26	94.51	88.53
	Rul*	8982	417	522	95.56	94.51	95.03
	Ent	9406	210	98	97.82	98.97	98.39
Test	***	861	215	0	80.00	100.00	88.88
	Dir	747	19	114	97.52	86.76	91.83
	Dir*	747	8	114	98.94	86.76	92.45
	Rul	826	164	35	83.43	95.93	89.25
	Rul*	826	37	35	95.71	95.93	95.82
	Ent	845	12	16	98.60	98.14	98.37

Table 1: Disambiguation results for various system configurations. Column and row headings are explained in the text.

The results for the directly comparable systems Dir*), Rul*), and Entr) show that the rule-based system is superior to the direct implementation; the statistical model is superior to both non-statistical systems. We achieve an overall recall of 98.14% as the best result on the test set. With a corresponding accuracy of 97.39%, the performance of the model we trained is comparable to the portable system Reynar and Ratnaparkhi used, given the fact that our training corpus consisted of 9500 sentences. However, we use a larger set of potential boundary characters.

Conclusion

We have presented a comparison of three different approaches to sentence boundary disambiguation: a direct method, a rule-based system, and a statistically trained model. Judging from the recognition results alone, the maximum-entropy system offers the best overall performance, yielding a recall of 98.14%, disambiguating six different potential boundary characters, and using a training data set with under 10,000 sentences. But even if the differences in accuracy were not significant, the trained system offers considerable advantages over the other approaches.

The development time for a new language is very short, initial results can be obtained within a few days by securing a small corpus of documents from the web and annotating them. Moreover, the selection and addition of features is straightforward using the perl-module approach for their implementation we have chosen.

The maximum-entropy system also has the advantage of being adaptive. During the development of other knowledge sources for translation (e.g. syntactic grammars), we need to constantly analyze sample sentences. These sentences are derived from further corpora we obtain. As the analysis proceeds, these disambiguated sentences can be used to augment the training data for the estimation of the model parameters.

In the future, we will experiment with an extended set of sentence boundary characters. Especially line breaks are of interest to us, since plain text documents often use them to partition documents. Additionally, the processing of quotes is important to segment embedded quotations correctly.

References

Darroch, J.N. & Ratcliff, D. (1972). Generalized Iterative Scaling for Log-Linear Models. *The Annals of Mathematical Statistics*, **43**(5), 1470-1480.

Grefenstette, G. & Tapanainen, P. (1994). What is a Word, What is a Sentence? Problems of Tokenization. In *Proceedings of COMPLEX 1994*, pp.7-10, Budapest, Hungary.

Mikheev, A. (1999). A Knowledge-Free Method for Capitalized Word Disambiguation. In *Proceedings of the ACL99*, pp. 159-168. Maryland, VA.

Mikheev, A. (2000). Tagging Sentence Boundaries. In *Proceedings of the NAACL*, pp 264-271, Seattle, WA.

Palmer, D.D. & Hearst, M.A. (1997). Adaptive Multilingual Sentence Boundary Disambiguation. *Computation Linguistics*, **23**(2), 241-269.

Ratnaparkhi, A (1997). A Simple introduction to maximum Entropy Models for Natural Language Processing. IRCS Report 97-08, University of Pennsylvania, Philadelphia, PA.

Reynar, J.C. & Ratnaparkhi, A. (1997). A Maximum Entropy Approach to Identifying Sentence Boundaries. In *Proceedings of the ANLP97*, Washington, D.C.

[1] Due to a technical problem, the systems Dir) and Dir*) could not process the complete training set. The figures given here reflect results with 7460 sentences.