Space Characters in Chinese Semi-structured Texts

Rongzhou Shen
School of Informatics
University of Edinburgh
rshen@inf.ed.ac.uk

Claire Grover
School of Informatics
University of Edinburgh
grover@inf.ed.ac.uk

Ewan Klein
School of Informatics
University of Edinburgh
ewan@inf.ed.ac.uk

Abstract

Space characters can have an important role in disambiguating text. However, few, if any, Chinese information extraction systems make full use of space characters. However, it seems that treatment of space characters is necessary, especially in cases of extracting information from semi-structured documents. This investigation aims to address the importance of space characters in Chinese information extraction by parsing some semi-structured documents with two similar grammars - one with treatment for space characters, the other ignoring it. This paper also introduces two post processing filters to further improve treatment of space characters. Results show that the grammar that takes account of spaces clearly out-performs the one that ignores them, and so concludes that space characters can play a useful role in information extraction.

1 Introduction

It is well known that a snippet of text in Chinese (or some other oriental languages) consists of a span of continuous characters without delimiting white spaces to identify words. Therefore, most parsing systems do not make full use of space characters when parsing. Furthermore, even though Latin-based languages such as English have delimiting white spaces between words, most systems treat them as no more than delimiting characters. Therefore, space characters are usually stripped out of the text before processing.

However, this is intuitively wrong. (Rus and Summers, 1994) stated that *"the non-textual content of documents complement the textual content and should play an equal role"*. This paper shows that space character plays an equal role as the textual content, where it can be used not only to construct a certain layout, but also to signal a certain syntactic structure. Some researchers have been seen to make use of space characters, but they mainly use spaces to create or recognise certain special layouts. For example, (Rus and Summers, 1994) used white spaces to reformat documents into somewhat structured styles; (Ng et al., 1999) and (Hurst and Nasukawa, 2000) used spaces to recognise tables in free text. Wrapper generation is more related to our research since it uses layout to extract structured content from documents (Irmak and Suel, 2006; Chen et al., 2003). However, wrapper generation is too high level, this paper is aimed at exploring the effects of space characters at a lower level.

In this paper, we focus on semi-structured documents (in our case, real-world Chinese Curricula Vitae), since these types of documents tend to contain more space layout information. This paper is intended to address the importance of space characters not only in layout extraction, but also in information extraction. To do so, Daxtra Technologies' grammar formalism and their additional elements for basic space character treatment is introduced [1]. Then an improved treatment plan is given for further disambiguation. Finally, we perform evaluation of the tools on a set of real-world CVs and give proposals for future work.

2 Space Characters

A space character, when considered as punctuation, is a blank area devoid of content, serving to separate words, letters, numbers and other punctuation. (Jones, 1994) found broadly three types of punctuation marks: delimiting, separating and disambiguating. Similarly, space characters have three different functionalities: delimiting, structuring and disambiguating.

Space characters are natural delimiters in some languages. In English and many other Latin-based languages for example, spaces are used for separating words and certain punctuation marks (e.g. period and colon). However, in formal Chinese typesetting, spaces are not used to delimit words or characters. Hence the need for automatic word segmentation systems (Zhang et al., 2003). The current segmentation systems mainly focus on resolving ambiguities and detecting new words in segmenting text with *no* spaces (Gao et al., 2005). However, ambiguities can be caused not only by characters themselves, but also the spaces and layout around them. The paper will later demonstrate this in terms of recognising entities, but the same should apply to segmentation.

Therefore, Chinese documents can have white spaces, it is up to the author of the document to decide when to use spaces, which makes dealing with people's spacing habits one of the reasons to include treatment of space characters in linguistic systems.

Structuring refers to space characters being used for layout purposes. For example, spaces and tabs can be used to create tables, putting spaces in front of a piece of text means to start a new paragraph etc. In some cases, such structuring space characters represent a relation between the elements that the spaces are delimiting. For the following example, each line contains a label and a value separated using spaces to create a table.

姓名(Name)	李某某
年龄(Age)	25岁
Email	li25@gmail.com
籍贯(Place of Birth)	上海

Disambiguating spaces occur where an unintentional ambiguity could result if the spaces were not there. Two types of ambiguities are usually caused by ignoring the effect of white space:

Overlapping Ambiguity, where a set of tokens can either be appended to the previous set of tokens to form an entity, or precede the next set of tokens to form a different entity. For example, in a Chinese CV's job history section, the following two situations could occur:

1999年10月1	日本公司会计
1999.10.1	A Japanese Company Accountant
1999年10月1日	本公司会计
1999.10.1	Accountant in this company

In the above example, two spans of text use exactly the same set of characters, but since the space is not in the same place, they have different meanings. Thus ignoring white space in this case could result in an overlapping ambiguity.

Combinatorial Ambiguity, where two sets of tokens can either be joined to form a single entity, or be separated to form two different entities. For example, "经理␣助理" could mean Manager Assistant when joined together, or since there are spaces in between the two words "经理" and "助理", they could also mean Manager and Assistant.

[1] Daxtra Technologies provides software for resume/CV parsing and extraction for candidate acquisition: http://www.daxtra.com

3 Basic Space Character Treatment

Daxtra Technologies' parsing system is a grammar formalism used to develop grammatical rules for recognising Named Entities and Relations. The system is based on context free grammar, but includes additional elements for integrating linguistic information (e.g. grammar and lexicon) and layout information (e.g. space characters) to parse structured and unstructured text. Along with parsing the text, the parser also labels the matched text with XML tags.

A typical Daxtra grammar rule looks like the following:

```
: person =
      person-firstname
    + person-lastname !ATTACHED_L
: person =
      person-firstname
    + person-midname !ATTACHED_L
    + person-lastname !ATTACHED_L
```

As the above example illustrates, a rule begins with a colon and the rule's name. For example, consider the following two person names:

Rongzhou Shen
Andrew Peter Baker

assuming that "Rongzhou Shen" matches the first rule and "Andrew Peter Baker" matches the second rule, then both will be surrouned by <person> XML tags.

Contents following the equal sign are a combination of other defined grammar rule names or lexicon names to build up the body of the `person` rules. Thus, for the first `person` rule to match a piece of text, the sub contents of the text must match `person-firstname` and `person-lastname` in the order given. Any other contents between a right hand side rule name and its XML tag replacement (i.e. the square bracketed contents) are attributes attached to the rule. These attributes include layout information.

For describing layout information, the Daxtra grammar formalism offers three types of space grammar rule: `ATTACHED` (`ATTACHED_L`, `ATTACHED_R`), `TABULATION` (`TABULATION_L`, `TABULATION_M`, `TABULATION_R`) and `LINEBREAK` (`LINEBREAK_L`, `LINEBREAK_M`, `LINEBREAK_R`).

ATTACHED This attribute checks the matching contents of the attached rule for surrounding spaces. Accordingly, `ATTACHED_L` detects spaces on the left of the matching contents, and `ATTACHED_R` detects spaces on the right of the matching contents.

TABULATION Similar to `ATTACHED`, this checks for tabulation characters in the matching contents. `TABULATION_L`, `TABULATION_M` and `TABULATION_R` checks for tabulations before, inside or after the matching text respectively. A tabulation is either a tab character or a span of more than three continuous white spaces.

LINEBREAK As the name suggests, this attribute checks for line breaks in the matching text. `LINEBREAK_L`, `LINEBREAK_M` and `LINEBREAK_R` checks for line breaks before, inside or after the matching text respectively.

4 Improved Algorithm

Although the Daxtra grammar formalism offers a full range of space layout descriptors, questions still arise. Consider the job history examples in Table 1. The first one would parse correctly with some simple grammar such as the following (assuming that we have all the needed lexicons):

```
: history = date-range !ATTACHED_R
          + company !ATTACHED_R
          + occupation !ATTACHED_R
          + occupation
```

However, the same rule would become ambiguous for the second example, where there is a space

Original	1999 - 2000	3CR Health Beauty International Ltd.	助理经理␣助理会计
Translation	1999 - 2000	3CR Health Beauty International Ltd.	Assis. Manager␣Assis. Accountant
Original	1995 - 1997	EMaiMai.com Hong Kong Ltd.	经理␣助理␣␣会计
Translation	1995 - 1997	EMaiMai.com Hong Kong Ltd.	Manager␣Assistant␣␣Accountant

Table 1: An example job history section in a CV file

between "经理" (Manager) and "助理" (Assistant). In such a case, two matches are found, as shown in Table 2.

We may notice that the word "助理" (Assistant) is closer to the word "经理" (Manager) than the word "会计" (Accountant), hence the correct entities being "经理助理" (Manager Assistant) and "会计" (Accountant). If on the other hand, there were more spaces between "经理" (Manager) and "助理" (Assistant) than "助理" (Assistant) and "会计" (Accountant), we may infer that the entities would be "经理" (Manager) and "助理会计" (Assistant Accountant).

Therefore, more control is needed for incorporating space layout information. For example, the problem in Table 2 can be resolved by comparing the number of spaces between the words. To do so, we replaced the spaces with XML tags with an attribute indicating the number of spaces replaced. For example, a span of four spaces will become: `<w spaces='4' />`. Based on such a transformation, we came up with the following post-processing filters for resolving ambiguities and other errors caused by space characters:

Filter *least-space* For different matches of the same rule, always choose the match that has the least number of spaces *inside* the entities.

For example, consider the two cases in Table 3. They both have exactly the same set of characters, but are in fact two different combinations, as indicated by the translations in the table.

Assuming that a simple rule like the following is used to match both the job histories:

```
: history  = date !ATTACHED_R
              + company
              + occupation
```

Then for (1) in Table 3, the two possible matches are shown in Table 4.

Therefore, the first match yields a total of one space inside the entities (between "年" and "冬"), while the second match yields three spaces (between "冬" and "宝洁公司"). Thus the first match is chosen.

Similarly for (2) in Table 3, there are two possible matches (see Table 5), in which the first has four spaces inside the entities and the second has two spaces, so the system chooses the second match.

Filter *equal-space* For a parsing with only one possible match, check whether the entity contains an unequal number of spaces between characters.

For example, "中国会计准则␣␣上市公司" (Chinese Accountant Regulations␣␣Listed Company) can be recognised by the system as a `company` entity, but it is in fact not. Thus in this case, the filter *equal-space* will reject it - there are no spaces between the first six characters, but two spaces appear after them, so the two spaces are not considered as part of an entity.

5 Evaluation

The evaluation data is a set of entities extracted from 314 real world CVs. The original CVs were all MS Word files, then converted to plain text using

Original	1995 - 1997	EMaiMai.com Hong Kong Ltd.	经理␣助理	会计
Translation	1995 - 1997	EMaiMai.com Hong Kong Ltd.	Manager Assistant	Accountant
Match 1	1995 - 1997	EMaiMai.com Hong Kong Ltd.	经理	助理会计
Translation	1995 - 1997	EMaiMai.com Hong Kong Ltd.	Manager	Assistant Accountant
Match 2	1995 - 1997	EMaiMai.com Hong Kong Ltd.	经理助理	会计
Translation	1995 - 1997	EMaiMai.com Hong Kong Ltd.	Manager Assistant	Accountant

Table 2: The second example's two matching variants

Original (1)	2002年␣冬␣␣␣宝洁公司全球会计和财务
Translated	2002␣Winter␣␣␣P&G Global Accountant and Finance
Original (2)	2002年␣␣␣冬␣宝␣洁␣公司全球会计和财务
Translated	2002␣␣␣Dong␣Bao␣Jie␣Company Global Accountant and Finance

Table 3: Examples showing two different combinations using the same set of characters. (Note: "DongBao-Jie" and "Winter P&G" have the same characters in Chinese.

Match 1	2002年␣冬	␣␣␣	宝洁公司	全球会计和财务
Translation	2002␣Winter	␣␣␣	P&G Company	Global Accountant and Finance
Match 2	2002年	␣	冬␣␣␣宝洁公司	全球会计和财务
Translation	2002	␣	Dong␣␣␣BaoJie Company	Global Accountant and Finance

Table 4: Two possible matches of (1) in Table 3

Match 1	2002年␣␣␣冬	␣	宝␣洁公司	全球会计和财务
Translation	2002␣␣␣Winter	␣	P␣&G Company	Global Accountant and Finance
Match 2	2002年	␣␣␣	冬␣宝␣洁公司	全球会计和财务
Translation	2002	␣␣␣	Dong␣Bao␣Jie Company	Global Accountant and Finance

Table 5: Two possible matches of (2) in Table 3

wvWare [2]. The converted files were all encoded using UTF-8. To demonstrate generality of the rules and filters, the selected CVs included differents kinds of layout, among which plain paragraphs, tables and lists are the most common. Table 6 shows the types of entities extracted.

To evaluate the effect of the different treatments of space characters, four sets of data were prepared, Table 7 shows the list of data.

For annotating the gold set, we performed named entity recognition using the latest grammar rules, then hand corrected the mistakes to produce a gold data set. For evaluation method, we used the standard Precision/Recall/F-score measures. To compute the standard measures, the XML output from the original parsed texts are converted to a CoNLL style format. For the example in Table 8, the converted CoNLL format looks like Figure 1.

5.1 The Results

A total of 24,434 entities were annotated in the gold set, Table 9 shows the distribution of the entity types among the whole set of entities.

After running each version of the grammar (i.e. Baseline, Version 1, Version 2) on the whole set of

[2] wvWare is an opensource project for accessing and converting MS Word files: http://wvware.sourceforge.net/

	Number of correctly labeled characters	Number of gold annotated characters	Number of system annotated characters	Precision (%)	Recall (%)	F1 (%)
Baseline	272901	302491	321059	85.00	90.22	87.53
Version 1	285736	302491	305339	93.58	94.46	94.02
Version 2	287365	302491	303948	94.54	95.00	94.77

Table 10: Results of each version computed against the gold data set

Entity Type	**Examples**
date	1990年10月1日, 1990.10.01
date-range	1998/10/1 - 1999/10/1
company	中信实业银行上海分行(Zhongxin Industrial Bank Shanghai Branch)
occupation	会计(Accountant), 经理助理(Manager Assistant)
person	沈容舟(Shen Rongzhou)
educational	爱丁堡大学(University of Edinburgh), 国防科大(University of National Defenses)
degree	学士(Bachelor), 硕士学位(Masters Degree)
subject	物理(Physics), 物理化学(Physical Chemistry)

Table 6: Types of named entities extracted from CVs.

Data name	**Description**
Gold	Human annotated data
Baseline	Daxtra grammar without space attributes
Version 1	Daxtra grammar with space attributes
Version 2	Daxtra grammar with space attributes and **least-space** filter and **equal-space** filter.

Table 7: The four sets of data prepared

Text	冬␣宝␣洁公司␣全球会计和财务
Translation	Dong␣Bao␣Jie Company␣Global Accountant and Finance
Rule	: history = company + occupation

Table 8: A sample text and its matching rule

```
冬  B-company
#space  I-company
宝  I-company
#space  I-company
洁  I-company
公  I-company
司  I-company
#space  O
全  B-occupation
球  I-occupation
会  I-occupation
计  I-occupation
和  I-occupation
财  I-occupation
务  I-occupation
```

Figure 1: The converted CoNLL style format for Table 8

CVs and converting the XML output into CoNLL format, there were a total of four sets of result files (including Gold annotated data set) and 1256 result files in total (one result file per CV). We then performed pair-wise comparisons of the result files from each version with the result files in the gold data set. Table 10 shows the final results.

As can be seen from Table 10, Version 1 is a great improvement over the Baseline in that both F1 score and precision increased by over 6%, while recall rose by 4.24%. This strongly indicates that the importance of space layout information is not to be

Entity Type	**Total Number**
date	10006
date-range	166
company	5456
occupation	3993
person	783
educational	1686
degree	1039
subject	1305
Total	24434

Table 9: Distribution of entity types in the CVs

neglected in named entity recognition tasks. A much lower number of system annotated characters for Version 1 shows that the layout information is disambiguating multiple matches, thus rejecting many predictions.

Although not as significant, Version 2 has still gained an improvement on performance over Version 1 by 0.75% in F1 score. A lower number of predicted annotations and a higher number of correctly predicted annotations both indicate more ambiguities have been resolved as a result.

Further investigations into the errors made by the Baseline showed that most ambiguities were overlapping ambiguities (over 90%). A possible reason for the smaller number of combinatorial ambiguities could be that people tend to be careful in writing their CVs, and tend to disambiguate entities by themselves. For example, instead of writing "经理 助理", separating the two words using a space, people will use punctuation marks to divide them. Furthermore, the case where people put spaces between each character wasn't so often seen: there were 16 CVs in total where such a case was found. Thus filter *equal-space* did not disambiguate many.

Further dividing the results down into smaller parts, we found that most of the ambiguities in the Baseline came from *company*, *educational*, *occupation* and *subject* names. This has two main causes: (1) These entities' grammar contain many generative rules, so ambiguities can not be avoided; (2) The context around these entities contain the most layout information (e.g. job history, educational history). Date and date-range entities were not affected so much by the layout information since they are straightforward to recognise. However, there was one case where the Baseline predicted a date wrongly:

> 1995年1月1日～1997年1月1　　日本公司樱花银行上海分行
>
> 1995.1.1 - 1997.1.1　　Japan Sakura Bank Shanghai Branch

The Baseline version predicted "1997年1月1日" as a single entity of type *date*. This is obviously a human typing error, where the author missed out "日" on the end of the date. This error was later fixed by Version 1.

From to the above discussion, we may know that *least-space* is mainly targeted at resolving overlapping ambiguities (which account for more than 90% of the ambiguities found), thus making it the more significant filter of the two.

Although Version 1 and Version 2 both had improvements over the Baseline, many errors still occur and they are categorized as follows:

- Rejections caused by filter *equal-space* were in fact real entities, uneven spaces in the entities were mostly human typing error;
- Choices made by filter *least-space* were occasionally wrong. This happens most often when two matches have a very small difference in the number of spaces inside entities;
- Grammars either overgenerate (cause plain tokens to be predicted as entities) or undergenerate (cause entities to be not detected);
- Lack of lexicon.

6 Conclusion

This paper has attempted to address the importance of space characters in Chinese linguistic parsing or information extraction in semi-structured documents. Essentially, space characters can contribute to the syntactic structure of texts and should not be only treated as delimiters or be stripped out of the document. This is especially true for semi-structured documents such as CVs.

As our results indicate, integrating simple layout information with linguistic grammars can greatly improve the performance of information extraction. A further improvement can be achieved using the two filters introduced in the fourth section.

Although Daxtra's grammar formalism is chosen as the tool for information extraction, since it already includes treatment of space characters, other tools are also available to carry out the same job. For example, Edinburgh University Language Technology Group's LT-TTT2 (Grover and Tobin, 2006) [3].

Our paper focuses mainly on Chinese CVs, but space layout information can be used widely in other languages and documents. In English for example, although words are separated by a single space, spaces are not always used as delimiters (e.g. constructing tables, columns), thus providing the need for integrating space layout. In terms of document types, plain paragraph based text (e.g. articles, blogs etc.) may not be affected too much by space characters, but integrating space layout information in parsing these documents should not decrease performance either. Furthermore, semi-structured documents may not be just limited to CVs: people's online portfolios, advertisements etc. all have space layout information attached. Therefore, much investigation still needs to be done on the effect of space characters in different types of documents.

References

Chen, Liangyou, Hasan M. Jamil, and Nan Wang. 2003. Automatic wrapper generation for semi-structured biological data based on table structure identification. *Database and Expert Systems Applications, International Workshop on*, 0:55.

Gao, Jiangfeng, Mu Li, Andi Wu, and Chang-Ning Huang. 2005. Chinese Word Segmentation and Named Entity Recognition: A Pragmatic Approach. *Computational Linguistics*, 31(4):531 – 574.

Grover, Claire and Richard Tobin. 2006. Rule-based chunking and reusability. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy.

Hurst, Matthew and Tetsuya Nasukawa. 2000. Layout and Language: Integrating Spatial and Linguistic Knowledge for Layout Understanding Tasks. In *Proceedings of COLING*, pages 334 – 340.

Irmak, Utku and Torsten Suel. 2006. Interactive Wrapper Generation with Minimal User Effort. In *Proceedings of the 15th International Conference on World Wide Web*, pages 553 – 563.

Jones, Bernard. 1994. Exploring The Role of Punctuation in Parsing Natural Text. In *Proceedings of 15th Conference on Computational Linguistics*, pages 421 – 425.

Ng, Hwee Tou, Chung Yong Lim, and Jessica Li Teng Koo. 1999. Learning to Recognize Tables in Free Text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 443 – 450.

Rus, Daniela and Kristen Summers. 1994. Using White Space for Automated Document Structuring. Technical Report TR94-1452, Cornell University, Department of Computer Science, July.

Zhang, Hua-Ping, Hong-Kui Yu, De-Yi Xiong, and Qun Liu. 2003. HMM-based Chinese Lexical Analyzer ICTCLAS. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 184 – 187.

[3] http://www.ltg.ed.ac.uk/software/lt-ttt2