Identifying and Tagging Titles in Web Texts

Clémentine Adam, Estelle Delpech, Patrick Saint Dizier

Identifying and Tagging Titles in Web Texts

Clémentine Adam, Estelle Delpech
IRIT-UPS
118 route de Narbonne
31062 Toulouse cedex France
clementine.adam@gmail.com,
delpech_estelle@yahoo.fr

Patrick Saint-Dizier
IRIT-CNRS
118, route de Narbonne
31062 Toulouse cedex France
stdizier@irit.fr

ABSTRACT

In this paper, we present an analysis based on linguistic and typographic features that allows for the identification of titles in web documents. We focus in particular on procedural texts. Identifying texts is a difficult task because ways pf encoding them are very diverse. A number of titles are also incomplete because fo context, we propose also a way to retrieve the missing elements, in particular predicates, so that titles are fully intelligible.

Keywords

structure analysis, text semantics, text titles

1. INTRODUCTION

Recognizing and tagging titles in web documents is a difficult but necessary task. Indeed titles are realized in a large number of ways which do not follow in most cases the standards. Titles found in Web texts may also have different roles: some are related to the page main contents, whereas others deal with external considerations such as advertising, links to blogs, hints and advices of various kinds, just to cite a few. In this project, we are basically interested in identifying titles which are related to the document contents. For that purpose, we will consider both surface (e.g. typographical) and contents marks.

Titles in text play a large variety of roles. They obviously structure documents [6], outlining the main topics addressed. They can be viewed also as denoting goals, as in procedural texts, the area we are concerned with here. Question answering systems, for How-to questions need to refer to this latter type of title, whereas tutoring systems need to refer quite accurately to both types. We will not address here the complex roles titles may play, but this would certainly be a very useful investigation. To our knowledge, little work has been done on tagging titles, besides what has been elaborated in the Text Encoding Initiative. Titles have been studied in psycholinguistic circles, in linguistics [4, 6] and in didactics.

Web texts produced by non-professionals do not have in general a very strict encoding in html. We conducted our investigations on procedural texts [1, 3] which include a large variety of domains and document types, including documents as diverse as teaching texts, medical notices, social behavior recommendations, directions for use, assembly notices, do-it-yourself notices, itinerary guides, advice texts, cooking recipes and video games solutions. In those documents we noted a large variety of titles: titles related to the text contents, but also a number of other elements encoded as titles such as adds, email addresses, hyperlinks, web navigation instructions, etc. The next problem is that titles are often incomplete, up to 65% in some procedural texts such as cooking recipes. It is obviously of much interest to be able to reconstruct those titles for an optimal use of those texts by users. Another illustration of this need is, for example, in a how-to question answering context, questions (How to change my mothercard ?) that have to match with a comprehensive title, where the response is the procedure that follows. If we go further, in some areas, it may even be useful to be add titles when sections are very long, however reconstructing (or inferring) missing titles is a different issue that may need techniques such as e.g. text tiling.

In this short document, we first address the issue of title identification and tagging in a large variety of types of procedural texts, next we show how titles can be reconstructed via a learning mechanism and how index can be added to titles to allow for question matching. Titles express indeed goals 'changing a bike wheel' which must match with how to questions 'how to change my bike wheel?' to get the answer which is the text that follows, a set of instructions.

2. RECOGNIZING AND TAGGING TITLES

2.1 Cleaning Web texts

The input of our system are raw Web pages. To be able to correctly tag titles, it is necessary to eliminate useless information (advertising, summaries, links to blogs, comments, etc.). This useless information can represent up to 66% of the text. To carry this out, we need

1. to extract relevant text, that is, any kind of text that is not navigation help, advertisements or comments posted by cybernauts and

2. to select and to simplify the html tags in so as to keep the main typo-dispositional information (para-

graph breaks, subdivisions of paragraphs into lines, lists and their subdivision into elements, emphasis).

Although (2) was quite an easy task, we had some difficulties achieving (1). We designed an algorithm that returns, for each paragraph, if its contents can be considered as relevant or not. It mainly uses paragraph length and proportion of closed-class words criteria. We evaluated it on 100 Web pages, from 12 different web sites. The results compared to a manual treatment are quite good, we have 0,95 precision and 0,76 recall.

When the text is 'clean', we apply the Treetagger on it to identify its morpho-syntactic terms. We just keep some categories of interest to us (e.g. verbs, connectors). We also make some revisions since, in French, the imperative form, which is central to our system of extraction patterns, is often identified as present indicative tense.

2.2 Recognizing Titles

For answering How-to questions it is obviously of much importance to recognize titles, which, in fact, mostly express goals of various levels. A second challenge is to possibly identify title hierarchies in complex or long texts. Automatically identifying titles is quite challenging and has been seldom addressed in the past. Obviously criteria depends on the type of text (pdf, word, html, etc.), the quality of the encoding, the type of text (procedural, roman, news, etc.) and the domain at stake.

Let us concentrate here on procedural texts, encoded in html format, from various sources, styles and domains. As advocated above, a problem for us is that a number of titles in web pages are irrelevant with respect to the procedure at stake, they are rather advertising, web services ('click here for more') or summaries, to cite just a few. Besides recognizing titles as such, our task is in fact to concentrate on titles related to a procedure, so that these can be used for answering questions.

Titles are short text sequences, highlighted (bold, color, underlined, large size or different type of font, etc.). A first observation is that html encodings are, by far, not homogeneous. Titles are coded with the tag $< h_n >$ in only 20% of the cases over the 600 titles observed. In most cases, the tag $< b >$ is used, possibly also $< emp >$, $< u >$ and a few others (macros...). Low level titles even have more unexpected encodings. Encodings may be quite homogeneous within a given web site, but heterogeneity prevails over different sites, even in the same domain.

To be more precise, we observed that, roughly:

- 80% of titles are encoded with $< b >$
- 57% of the total of $< b >$ used in texts encode titles
- 64% of the total of $< h >$ used in texts encode titles.

This means that we need to consider additional criteria, among which:

- typography (spacing w.r.t. paragraphs before and after),
- the contents (number of words, inflected verbs) in the segment assumed to be a title,
- the type of elements after the title (e.g. instructions, which are a good indicator of a procedural title).

Titles are identified in two steps. First, an algorithm traverses paragraphs of a text one by one, and assigns them one of the following tags: `title, text` or `ambiguous`. This first step is quite straightforward. From our investigations on procedural texts, a title is a paragraph composed of a sequence of words of less than 12 words long and bearing emphasis. The tag `text` will be assigned without any doubt if the paragaph is subdivided into smaller units or is longer than 12 words. Ambiguous parapraphs are mainly short sequences of words (12 words or less) with no emphasis.

The second step disambiguates the ambiguous paragraphs one by one, using the tags assigned by the first step to their surrounding paragraphs. For example, an ambiguous paragraph between two paragraphs tagged as `text` will be considered as a `text`. Similarly, we have the following rules:
'an ambiguous paragraph between two titles is a text',
'an ambiguous paragraph followed by a title becomes a text'
'an ambigous paragraph becomes a title if it is the first paragraph of the text', etc.

This second step also operates some repairs on the tags yielded by the first step. For example, any sequence of more than two titles, i.e : "title title title", will be changed to "title title text".

The title hierarchy is very difficult to identify without content analysis. In fact, it is often largely pragmatic in nature. For example in 'The pizza Margarita .... the paste .... the topings .... the serving ...'. It is impossible a priori to hierarchically organize those subtitles if you do not know what pizzas look like.

However, standard procedural texts are not very long and tend to be relatively linear. This means that, besides the page title, we observed in 80% of our texts not more than 2 levels of titles (exluding the main title). We observed two regular types of titles that can be correlated to some form of hierarchy. Type 1 is a title separated from the paragraph that follows by a $< p >$ tag. Type 2 is a title separated from the paragraph that follows by a $< br >$ tag. Although we still have no means to tell the exact level for titles, we can quite confidently say that a type 2 title will be at a lower level than a type 1 title, whatever the website or the domain. This information may be useful for question-title matching: type 2 titles are expected to introduce paragraphs that deal with more specific aspects of a procedure than paragraphs introduced by a type 1 title. Type 2 titles could help answering specific questions.

2.3 Evaluation

The evaluation corpus is composed of 78 Web pages over 5 domains: coocking recipes, do-it-yourself, video game solutions, social life, and medical recommendations. The total

number of words is 61159, this not very large, but we feel sufficient for an indicative evaluation, giving us directions to improve the system. For each sequence, two annotators, doing the same task, had to decide whether it is a title or not. The corpus contains 4560 sequences, among which 511 titles and 1641 sentences containing at least one instruction.

The title recognition algorithm yields the following results over 5 different domains. Precision was given priority over recall to avoid errors as much as possible. We report here the recognition of titles related to text contents.

domain	recall	precision	certainty
cooking receipes	0.72	1	0.83
do it Yourself	0.8	0.96	0.87
social life	0.69	0.97	0.80
video games	0.61	0.93	0.74
medical notices	0.58	0.81	0.67

2.4 Filtering out non-relevant titles

The next step is to filter out as much as possible titles which are not relevant w.r.t. the contents of the page. The number of useless titles may vary quite largely depending on the domain. In general, we observed about 20 to 25% of irrelevant titles.

For that purpose, we consider two techniques that we are investigating at the moment:

- define a 'stop list' of typical terms found titles which are not relevant (e.g. click, see, consult, confirm, buy, advice, recommendation, etc.). So far, with a short stop list of 163 words, 59% of irrelevant titles are filtered out and only 4 titles out of 276 informative titles have been erroneously filtered out. This list may clearly be extended since so far the noise introduced is marginal.
- keep titles that have common contents with the paragraphs that follow. In particular, we are evaluating the fact that a relevant title must contain words, or related terms (synonyms, holonyms) [2], that appear frequently in the paragraphs that follows. A technique based on text tiling can be used.

3. RECONSTRUCTING ELLIPTICAL TITLES

3.1 The problem and the situation

In procedural texts, there are domains like cooking recipes where we observed an average of 56% and up to 65% of the titles which are incomplete, i.e. w.r.t. the basic form: 'predicate + object argument' either the predicate or the argument is missing. For the reader, the reconstruction of the missing element is often straightforward due to context. Our goal is to identify the elements in the text that allow us to reconstruct titles, and possibly, as a side effect, to index them for information retrieval or question answering purposes. In general, elliptical situations do not really depend on the domain, there are however slightly more such titles in cooking recipes (references are probably more straightforward), and slightly less in the 'pratical life' domain. Also, the deeper the titles are in the hierarchy, the more elliptical they are (from 31% for top titles to 86% in average for the lower level titles). Finally, texts which have a large number of titles have a slightly higher rate of elliptical titles (ranging from 40% to 60% in average for texts with more than 6 titles).

3.2 missing argument

The case where subtitles have a missing object is relatively simple to resolve: in most cases, the object of the main title is inherited by the lower titles. In general, we observed that this form of inheritance only concerns those titles which are just one level below. This simple strategy has 94% accuracy.

3.3 missing predicate

The case where the predicate is missing is the most frequent and the most complex to resolve. An approach is to deploy a learning mechanism, where, roughly, we consider a sample of titles which are fully realized (predicate + argument). The principle is then to collect all the verbs that appear in the instructions below this title. Learning consist then, roughly, in making a distributional analysis of the verbs that appear under a certain title verb. For that purpose, we considered a development corpus of 3000 web texts over various domains. Via the TextCoop text tagger [3], those texts are annotated.

From each title which is complete, we created the structure: <verb of title> - [list + frequency of the verbs in the instructions under the scope of that title]. Then, summing over all texts and titles, we have a structure such as:
<verb of title + frequency> - [list of verbs + frequency],
where the frequency associated with the 'verb of title' is the number of times this verb has been found in titles and the 'list of verbs' is the union of all verbs encountered under the 'verb of title', with frequencies for each verb. This list is obviously dependent on the domains considered, or the group of domains, as in our case where closely related domains have been considered alltogether.

We then constructed the inverse list, where an entry is a verb from the 'list of verbs':
<verb in instructions, frequency> - [list of title verbs associated, with frequencies].
This inverse list is used to reconstruct missing verbs in titles.

Then, to reconstruct a verb in a title with no verb, we proceed as follows. Given a title with a missing verb, we construct the list of verbs in the instructions in the scope of that title. From the inverse list above, we select potential title verbs, construct the union + frequencies set and, finally, keep the three most prominent verbs (those with the highest frequency) or deverbals.

This simple approach gives the following rates, via a manual analysis, considering again complete titles (but simulating lack of predicate), so that we have the solution accessible: in 48% of the cases, the correct verb has been proposed, and in 65% of the cases a good verb, closely related, has been proposed.

Now, we can pair this algorithm with an endogenous search: we search in the paragraphs below the title if the argument which is the title is used and combined with a verb. This happens in 29% of the texts. If we combine the two techniques, learning and endogenous search, then, the results are really satisfactory: the correct verb is proposed in the list

< *procedure* > < *titlelevel* = "0" *index* = "*embellish, paint, decorate*" > How to embellish your balcony < /*title* >
< *Prerequisites* > 1 lattice, window boxes, etc.< /*prerequisites* >
....
< *titlelevel* = "1" *index* = "*cleaning, sweep, wash*" *missing* − *arg* = "*balcony*" > Cleaning < /*title* >
..... (instructions).....
< *titlelevel* = "1", *index* = "*adding, including, decoratingwith*" *missing* − *verb* = "*adding*" > plants < /*title* >
..... (instructions).....
< *titlelevel* = "1", *index* = "*spreading, painting, choosing*" *missing* − *verb* = "*spreading*" > the paint < /*title* >
..... (instructions).....
.....
< /*procedure* >

Figure 1: Annotated titles in a procedure, (gloss from French)

in 62% of the cases and a closely related verb is proposed in 86% of the cases.

An interesting feature is that the verbs present in 90% of titles are not so different: we have about 52 recurrent verbs which are quite generic for the group of domains we considered: *choose, maintain, use, make, put, clean, paint, replace, prepare, manage, plant,* etc. We also have a small number of deverbals derived from those verbs (replacement, preparation, etc.). Obviously for domains like health or video games, this list would be notably different.

3.4 Indexing titles

Finally, we can, based on the list of proposed verbs and deverbals, index all titles (complete or not) by means of that list, for question answering purposes. Consequently, in our representation, any title recieves a list of closely related verbs and deverbals as indexes, which will be used when attempting to match the terms of an How-to question with a title. So, instead of searching quasi-synonyms [2] via a lexicon or an ontology as it is often the case in question matching procedures, resource which is not readily available for verbs in most domains, we have a list of predefined indexes which can be used directly with a good relevance score.

The representation of a title is as follows:
<title index="w1, w2, w3"> ... title .... < /title>
where w1, w2 and w3 are indexes, ranked by decreasing frequency.

4. PERSPECTIVES

In this short paper, we have presented the way titles in web pages, in a large variety of procedural texts, can be identified and tagged. We have also shown how to filter out titles which are not relevant w.r.t. the text contents. Finally, we have shown how to reconstruct titles which are elliptical, and how this reconstruction allows the production of dedicated indexes. These indexes are used for question matching since titles represent goals, similarly to questions.

In the near future, we need to evaluate the quality of the matching question-answer. Preliminary analysis seems to give quite good results. Also we want to provide the user with a few responses, not just one, so that he can choose the one that corresponds the best to his expectations.

The transposition of these techniques to other domains, such as news and technical documents does not seem so straightforward, but this is certainly one of our goals: to be able to identify titles and reconstruct them when incomplete in a number of situations. Another application, more oriented towards didactics and tutoring systems is to be able to suggest additional subtitles when a long text lacks titles.

Acknowledgements This work is funded by the French ANR programme, RNTL section. This is part of the TextCoop project. We thanks all project members who contribuetd to discussions around the problem of titles.

5. REFERENCES

[1] Aouladomar, F., Saint-dizier, P., *Towards Answering Procedural Questions*, Workshop KRAQ05, IJCAI05, Edinburgh, 2005.

[2] Cruse, A., lexical Semantics, Cambridge Univ. Press, 1986.

[3] Delpech, E., Saint-Dizier, P., Investigating the Structure of Procedural Texts for Answering How-to Questions, LREC 2008, Marrakech.

[4] Jacques, M.P., Approche en discours de la réduction des termes complexes dans les textes spécialisés, PhD dissertation, Univ. Toulouse 1, France, 2003.

[5] Kosseim, L., Lapalme, G., *Choosing Rhetorical Structures to Plan Instructional Texts*, Computational Intelligence, Blackwell, Boston, 2000.

[6] Vallduvi, E, Engdahl, E., "The linguistic realization of information packaging". Linguistics 34(3), pp. 459-519, 1996.