Enriching How-To Guides by Linking Actionable Phrases

Alexandr Chernov[*]
University of Tübingen
Wilhelmstr. 19
Tübingen, Germany
alexandr.chernov@uni-tuebingen.de

Nikolaos Lagos   Matthias Gallé   Ágnes Sándor

Xerox Research Centre Europe
6 chemin de Maupertuis
Meylan, France
firstname.lastname@xrce.xerox.com

ABSTRACT

The World Wide Web contains a large number of community created knowledge of instructional nature. Similarly, in a commercial setting, databases of instructions are used by customer-care providers to guide clients in the resolution of issues. Most of these instructions are expressed in natural language. Knowledge Bases including such information are valuable through the sum of their single entries. However, as each entry is created mostly independently, users (e.g. other community members) cannot take advantage of the accumulated knowledge that can be developed via the aggregation of **related** entries. In this paper we consider the problem of inter-linking Knowledge Base entries, in order to get relevant information from other parts of the Knowledge Base.

To achieve this, we propose to detect *actionable phrases* – text fragments that describe how to perform a certain action – and link them to other entries. The extraction method that we implement achieves an F-score of 67.35%. We also show that using actionable phrases results in better linking quality than using coarser-grained spans of text, as proposed in the literature. Besides the evaluation of both steps, we also include a detailed error analysis and release our annotation to the community.

Keywords

procedural knowledge; forum mining; blog analysis

[*] Work done while at Xerox Research Centre Europe.

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2890585.

1. INTRODUCTION

The World Wide Web contains a great quantity of community created procedural knowledge, expressed mainly as natural language instructions, on how to perform certain actions: e.g. how to chose a PC, how to install an application on a smartphone, or how to cook spaghetti. Some of the most popular websites in that space include: WikiHow[1], Snapguide[2], eHow[3], WonderHowTo[4], Instructables[5]. Most of them are written by enthusiastic, not-paid contributors, who, most frequently, do not take the time to check for relationships between the newly created content and previous entries in the system. Knowledge bases (KBs) that contain such content are valuable through the sum of their single entries, but because each entry is created mostly independently, users (e.g. other community members or even software accessing such information) can not take advantage of the accumulated knowledge that can be developed via the aggregation of **related** entries.

In commercial settings the scenario is not so different. Customer care departments managing KBs that contain troubleshooting and implementation how-to guides do not always follow rigorous processes for their creation. Business pressure and short iteration frames do not give time to reorganize and optimize periodically those KBs.

Even worse, what is "just" a loss of time in the current situation risks becoming a bottleneck in the not so distant setting where some software (e.g. intelligent personal assistants and so-called conversational agents such as Siri[6], Cortana[7], Alexa[8], the Xerox Virtual Agent[9]) handle troubleshooting sessions with little (or no) human supervision. Interlinking KB entries is a crucial pre-requisite in such cases, as it would

[1] http://www.wikihow.com

[2] https://snapguide.com

[3] http://www.ehow.com

[4] http://www.wonderhowto.com

[5] http://www.instructables.com

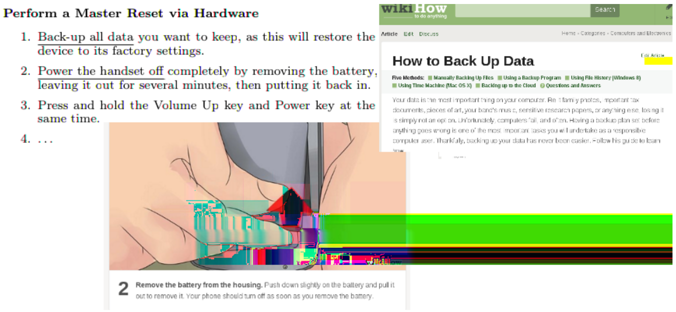
[6] http://www.apple.com/ios/siri/

[7] https://www.microsoft.com/en-us/windows/cortana

[8] https://developer.amazon.com/public/solutions/alexa/alexa-skills-kit

[9] http://www.wds.co/product/self-care/virtual-agent/

allow presenting instructions at different levels of detail (for instance for adapting the displayed content to the (inferred) expertise of the user). An example is given in Fig. 1, where the underlined spans correspond to instructions that are further developed in other parts of the KB.



Perform a Master Reset via Hardware

1. Back-up all data you want to keep, as this will restore the device to its factory settings.
2. Power the handset off completely by removing the battery, leaving it out for several minutes, then putting it back in.
3. Press and hold the Volume Up key and Power key at the same time.
4. . . .

wikiHow
to do anything
Search
Article
Edit
Discuss
Home » Categories » Computers and Electronics
Edit Article

How to Back Up Data

Five Methods: Manually Backing Up Files
Using a Backup Program
Using File History (Windows 8)
Using Time Machine (Mac OS X)
Backing up to the Cloud
Questions and Answers

Your data is the most important thing on your computer. Be it family photos, important tax documents, pieces of art, your band's music, sensitive research papers, or anything else, losing it is simply not an option. Unfortunately, computers fail, and often. Having a backup plan set before anything goes wrong is one of the most important tasks you will undertake as a responsible computer user. Thankfully, backing up your data has never been easier. Follow his guide to learn

2 **Remove the battery from the housing.** Push down slightly on the battery and pull it out to remove it. Your phone should turn off as soon as you remove the battery.

Figure 1: Example of a knowledge base entry and spans of text that link to other entries.

The settings we are focusing on are the following:

1. A human reader in a self-care environment, e.g. other members of the online community, who could click on the link (or where a short summary could be displayed when hovering over with the mouse) to know more details.

2. A software agent/intelligent personal assistant, who could use this linked information to guide a user through performing specific instructions.

In an idealistic setting we would like to establish a link between any span of text that refers to another entry in the KB. Of course, focusing on any possible span makes the search extremely large as the number of spans grows quadratic with the length of the text. In addition, most spans are uncorrelated with any entry. Instead, in this paper we focus on *actionable phrases*, assuming that those phrases could be further detailed to improve the understanding of a user and concentrate the highest potential for linking.

2. RELATED WORK

The growth of web forums has attracted lots of attention on methods that can aid in mining and organizing their content. However, the specific case of *procedural knowledge* (also called how-to knowledge) – defined as "the knowledge required to perform certain tasks" [10] – has received less attention.

[6] and [7] propose to extract procedural information based purely on structural properties of online forums, which typically contain: a title denoting the main task that the process achieves, and items of numbered or bullet lists representing a hierarchical structure of the steps involved in the process and the order in which these steps should be performed. After extracting the steps, a text search engine and a classifier are used to find a set of candidate links for each step and filter out irrelevant results. In that work, they suggest using a coarse-grained structure for the linking stage, that of a step (i.e. item of a numbered or bullet list), which may include several different actionable phrases. In our experimental work (Section. 4.2) we compare our results explicitly to this approach and show the advantage of using a finer-grained notion of a text span.

The specific problem of extracting verbs denoting actions and their direct objects has been addressed by [10] and [8], however they do not mention any effort of linking the extracted spans to other parts of a KB.

In addition, [9] does tackle the problem of linking entities extracted from how-to documents to a KB but this includes comparing the corresponding entity mentions to structured metadata (i.e. typically one or two token labels found in the KB) rather than unstructured text as in our case. In addition, they use a pre-defined lexicon of verbs to identify the actions mentioned in the text.

Note that in general what we are proposing is different from the well-studied problem of Named Entity extraction and linking (NEL) [5]. NEL does not consider procedural knowledge, meaning among other things that verbs are typically out of the task's focus. Furthermore, NEL systems link textual mentions to (semi-)structured knowledge bases rather than unstructured text.

3. METHOD

We define an *actionable phrase* as a text fragment that describes how to perform a certain action in an instruction. An actionable phrase includes the **action** itself (e.g. click, type) and what we call the **actee** i.e. the main object on which the action is performed and its distinguishing characteristics (e.g. spatial information). For instance, consider the sentence ”install drivers for a video card”, where ”install” represents the action, ”drivers” the direct object and ”for a video card” is part of the ”drivers” characteristics that can help us find a more precise result when the linking is performed.

As discussed in Sect. 1, our main focus is on the semantic elements that compose actionable information as well as their inter-relations. As a consequence, we limit the search for actionable phrases within the part of the documents that describe the sequence of actions carried out to achieve a specific objective, called procedures in this work. For instance ”Power the handset off” is in the scope of our definition with ”Power ... off” representing the action and ”the handset” the direct object. To identify such sections, we use the structure of the document. This is based on the hypothesis that how-to documents, in general, are well structured with the title stating the main problem/topic that the document covers while actionable information is covered by the content of items represented as numbered or bullet lists[10]. However, more advanced techniques could potentially be used.

Once the parts of the documents including actionable information have been identified, each point within the numbered and bullet lists, called in this work *step*, is further segmented into sentences.

Each sentence is then further analysed to identify actionable phrases. To achieve that, we have characterised the elements we want to extract as follows.

- **Actions**: They are represented by action-verbs. We used the following set of rules to identify such verbs:

    1. Verbs at the beginning of a sentence.

[10] This hypothesis has been verified by studying a number of different corpora, including WikiHow, Snapguide, and eHow, while in addition in the work of [6] and [7] a similar assumption is made.

2. Verbs following a modal verb.
3. Verbs which are the same as their infinitive.

- **Actees**: Words following an action (and standing before the next action) are candidates for representing the corresponding actee. In linguistic terms this includes the object of the action-verb and the longest linguistic expression in the same sentence whose global reference is that object.

    1. At a first stage, elements of the actee are selected based on their Part-of-Speech (POS) tag. The allowed tags include: nouns, adjectives, adverbs, and pronouns (determiners and prepositions are also allowed at that stage because they will be filtered out in the linking stage while having a continuous span of text to display helps when showing these spans to the users). An actee cannot end with a preposition, determiner, or pronoun.
    2. At a second stage the extracted objects are compared against a list of terms that represent domain-specific entities. If they match, then the corresponding actionable phrases are considered as candidates for linking. Such a list may be coming from an enterprise/application-specific terminology or external resources (e.g. Wikipedia). The longest (in number of tokens) fragment is selected and treated as the entity to be used for linking.

After identifying and extracting actionable phrases we search for relevant information in the rest of the KB. In our setting we used standard Information Retrieval (IR) techniques, searching over an index of the text of other documents found in the KB. However, other more advanced techniques could be used (for example a binary classifier, or learning-to-rank methods). The set of candidate links retrieved is then filtered using an experimentally chosen threshold.

4. EVALUATION

4.1 Experimental Setting

This section describes experiments for evaluating both the actionable phrase extraction and the corresponding linking.

4.1.1 Dataset

A publicly available dataset, which would be appropriate for our purposes, does not exist to our knowledge. In this Section we describe the corpus we used to perform our experiments and the gold standard annotation created for both the extraction and linking operations.

We obtained the data for these experiments from WikiHow, limiting our crawling to the forum posts classified under the category *Computers and Electronics*, as marked on WikiHow articles, and its subcategories: *Basic Computer Skills, Install-Uninstall Software, Maintenance and Repair, Phones and gadgets, Tablet Computers*. Overall we retrieved 1 758 articles.

Out of this corpus, we randomly selected 20 articles and asked two human annotators to create a gold standard against which the automatic extraction can be compared, by annotating the segments that represent an actionable phrase. We developed a custom annotation tool to annotate actionable phrases as correctly or incorrectly identified, mainly because we wanted to be able to cover the annotation of both actionable phrases and their links in the same environment[11]. This resulted in a total of 720 annotations. The agreement reported in this paper is based on these annotations. To compute recall, we also marked false negatives in the instruction text. The agreement between annotators was measured with the kappa score, obtaining a value of 0.779. As reported in the literature [4, 7], annotating text spans that indicate procedural information is a non-trivial task. We hope that the release of this dataset will enable further research in this area.

For each of the 720 annotations, we asked three human annotators to mark as correct or incorrect the results of the linking stage (the linking method is described in 4.1.3). This required reading through the corresponding documents to judge whether each of the results was correct. The position in the list of the results (i.e. top ranked result against second or third) was not taken into account. Generally the results were few (on average around three). We defined a threshold on the maximum number of top ranked results kept by our retrieval engine, set at 5. The agreement between the annotators in terms of kappa is shown in Table 1.

Table 1: Kappa scores between annotators for the linking of actionable phrases.

Annotators	**Kappa**
1-2	0.71
1-3	0.69
2-3	0.71

To be able to compare our results with the state-of-the-art method of [6] and [7], we asked the annotators to perform the same task for steps (i.e. full phrases corresponding to the text of each item in bullet and numbered lists). Obviously except for the difference between steps and actionable phrases, the same set-up as in the actionable phrase setting e.g. top 5 threshold, was kept. The agreement is shown in Table 2.

Table 2: Kappa scores between annotators for the linking of steps.

Annotators	**Kappa**
1-2	0.91
1-3	0.77
2-3	0.86

To allow further research in this subject we decided to release the annotated corpus, which can be accessed at `https://wic2016.xrce.xerox.com/corpus_WIC2016paper.zip`.

4.1.2 Implementation of actionable phrase extraction

Most of the articles crawled from WikiHow are very well structured. Each title corresponds to a specific problem/topic.

[11] Because of limited space and as the description of the tool is out of the scope of this paper we do not give more details here. More information can be provided upon request.

If a problem has more than one solutions, the article is divided into sections with informative subtitles. Instructions consist of steps, represented as items of numbered or bullet lists, as explained earlier in the paper. We used the WikiHow specific markup (Wiki-markup) to identify titles, subtitles, and steps.

Each step was segmented into sentences using NLTK [2].

For each sentence we performed Part-Of-Speech (POS) tagging using the Xerox Incremental Parser (XIP) [1]. To put together a domain-specific terminology, we enriched XIP's standard list of named entities (covering standard types such as organisations, people, etc.) using Wikipedia (version January 2015). The new list was built using the titles of Wikipedia articles from the (i) *Mobile Technology* and (ii) *Software* categories. Initially we got 33 954 titles. Entities that were unlikely to be nouns (based on POS-tagging information), as well as the titles containing file names were filtered out. After that we also removed the following words (based on a study of the corresponding list and manual post-processing): Open, INSERT, Make, Format, Start, Replace, A, Plug-in, SET, RUN, preview, switch, clean, clear, backup, type, visit, shutdown. Text inside brackets was removed. In the final list 33 708 entities were kept and were encoded as a Finite State Machine (FST) used for our experiments (the FST was integrated in the XIP framework).

The rules described in Sect. 3 were then used to identify the actions and actees forming actionable phrases.

4.1.3 Implementation of actionable phrase linking

An index of the entries was built using the Whoosh library[12]. We included in the index article and section titles, as well as the body text of each article. As titles are more concise and informative we assigned a higher weight to them rather than to the standard body text (3.0 and 1.0 respectively).

All documents containing at least one of the terms of the queries were scored using the BM25F ranking function (with default parameters) Only the top 5 candidate links were kept and further filtered out if they had a score equal or lower to 15 (the threshold was decided empirically).

As mentioned in 4.2.1, as a baseline we used the approach proposed in [6] and [7] which uses whole steps instead of actionable phrases. For comparison purposes, we kept the rest of the pipeline unchanged.

4.2 Results

4.2.1 Actionable phrase extraction

In our setting, detecting at least a part of the actionable phrase is important, as it allows to search for a link in the second stage of our approach. Based on that assumption, the average F_1 is 67.35, as shown in Table 3.

Table 3: Results of actionable phrase extraction

Annotator	Precision	Recall	F_1
1	72.20	63.23	67.72
2	77.00	56.93	66.97

If we take a stricter approach where a phrase is considered correct only if the exact boundaries of the annotations are found (closer to traditional information extraction), while a penalty is given if there is only a partial intersection of tokens between the annotations done by the human annotators and the text spans or mentions detected by our system (we fixed the penalty to 0.5 out of 1 in our case), the F_1 averages to 59:00.

The results indicate that, although the above step can be further improved, the corresponding implementation of the algorithm gives exploitable results (see Sect. 5.1 for a more detailed analysis).

It is very difficult to compare our extraction work to previous studies, as the same setting has not been considered before. A similar evaluation was done by [10] and [8] but on different corpora and with different types of targeted entities. Just to give an idea to the readers, [8] reports 77:60 of F_1 for what seems to correspond in our work to the action-verb plus the direct object of that verb. [10] reports 86:04 of F_1 for the action-verb, direct object and temporal, quantitative and spatial modifiers of the action-verb. None of the above authors performed linking.

4.2.2 Actionable phrase linking

We analyzed the annotations of the proposed links, which were labeled as being relevant or not to the selected span. As mentioned in previous sections, we used as baseline the algorithm using whole steps, in order to measure the impact of discovering finer grained phrases (i.e. actionable phrases). Fig. 2 shows the distribution of the ranking scores. The obtained scores when using steps are much more spread out, which is understandable as in general the text span is much longer. More importantly, the relevant group seems more separated from the irrelevant one in the actionable phrase scenario, while the two groups overlap much more when using whole steps.

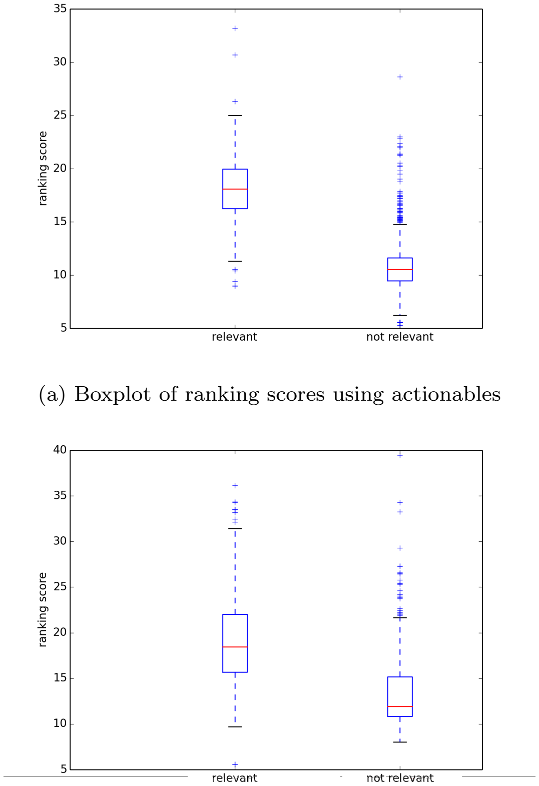
The conclusion that the separation between relevant and irrelevant text spans is more distinct in the case of actionable phrases than in the case of whole steps is confirmed by the ROC curves in Fig. 3, where the False Positives Rate (proportion of non-relevant documents that are retrieved, out of all non-relevant documents available) versus the True Positive Rate (Recall) is plotted. Clearly the ROC curve corresponding to actionable phrases dominates over the curve representing whole steps, resulting in a higher area under the curve (0:94 versus 0:82). Remember that the recall is computed based only on the links that are over a threshold fixed at 15. For actionable phrases there were 60% more such links found than the ones obtained when using steps.

5. DISCUSSION

5.1 Error Analysis

In this section we present issues we identified with the current extraction framework after performing an analysis of the results. We identified three main causes of error: the narrow definition (linguistically) of what an action is, the limited used of advanced syntactic analysis, and the fact that linguistic structures such as ellipsis and coordination are tough cases for standard NLP tools. This is in addition to errors introduced by specificities of the domain. For example in the following sentence the character >, is incorrectly interpreted as a sentence boundary, therefore missing the information that it actually represents a sequence of actions.

[12] https://pypi.python.org/pypi/Whoosh/



(a) Boxplot of ranking scores using actionables

(b) Boxplot of ranking scores using steps

Figure 2: Score discrimination capacity using actionable phrases and steps

Choose File > Reveal in Finder to see your picture files.

Such domain specific characteristics of non-standard text are a known hurdle for standard NLP tools [3].

5.1.1 Actionable phrases missed by the system

The rules implemented only consider actionable phrases where the actions are conveyed by a verb, either an imperative or following a modal verb. However, people use other linguistic structures as well for expressing the necessity of carrying out a task. For instance, consider the sentence,

make sure the "OK" button (is highlighted)

where even though there is an imperative verb, it is not an action. The action is instead *highlight*, which in that case is not covered by the rules defined in the method. Another interesting example can be found in the following sentence:

After typing the exact username ...

In that case, the action *type* is embedded into a subordinate clause, so the current system is going to miss it.

5.1.2 Wrongly identified actionable phrases

There are cases where the verbs extracted as actions correspond to the rules defined in the current method, however semantically they are not actions due to the semantic features of the verbs, such as in the following examples:

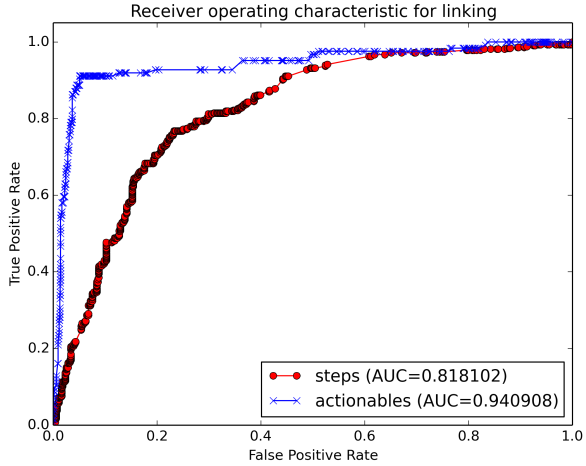


Figure 3: ROC Curve using the scores when querying with actionable phrases (blue) and steps (red).

you MUST know the exact username
You will see the Photo Booth application

A list of verbs denoting cognitive activity and perception needs to be defined, for instance based on a study of the corpus or based on external resources.

We also feel that the addition of advanced syntactic analysis in the information extraction system would help. For instance, sometimes the verbs come after an auxiliary but since their subject does not refer to the user (e.g. via the pronoun *you*), they do not convey an action, as currently indicated in the method. For instance:

it should flash windows

The use of an appropriate syntactic parser would filter out such noise.

5.1.3 Advanced linguistic structures

Linguistic structures like ellipsis, coordination or anaphoric expressions are hard cases for standard NLP tools, as illustrated in the following examples:

After typing the exact username, make sure the "OK" button is highlighted, if not, do so.
Shut down and restart a remote computer

For instance in the second sentence only *restart a remote computer* was detected by our algorithm while *Shut down* was missed, partly because of the inappropriate interpretation of the coordination denoted by *and*.

Deciding whether a prepositional phrase is a complement or a adjunct is also important when deciding if a prepositional phrase is an actee or not. Consider the following two examples:

Turn on a computer at your school.
type 'photo booth' in the search bar

In the first sentence, *at your school* was not considered as part of the actee - since it is an adjucnt -, while in the second case *in the search bar*, was noted as part of the actee - since it is a complement. The current method does not take into account this difference, which technically needs the integration of subcategorization information in the NLP tool.

Another question raised is if **possible** actions should be considered as actionable phrases or not, as in the example sentence below:

> *If you click on the button that shows a window on the bottom left corner, and then ...*

Those phrases can be enriched with a "condition" type (i.e. denoting a conditional statement), such as in [10].

5.2 Observations about the linking method

In our setting we have used standard Information Retrieval (IR) techniques, searching over an index including the text of other documents found in the KB.

A first improvement would be to link to sections of a document rather than the full document. As we are already able to perform the corresponding segmentation, we also plan to perform section-based indexing to validate our hypothesis.

Furthermore, the titles of such sections are very informative (in a similar manner to the titles of the documents). We believe therefore that processing in a customized manner different parts of the documents (for instance assigning different weights for specific parts of the document, as we have done for the titles) would be beneficial.

The distinct separation between relevant and irrelevant results observed in the linking experiments indicates that techniques such as binary classification, or learning-to-rank, could give us a better result than the experimentally selected threshold used in this study. This of course would require more annotated data, which we could use as seeds for training a classifier. Although the annotation process is time-consuming, and very often not trivial, statistical machine learning is still worth investigating. The same is true for the extraction component, although special attention should be given to the considerations mentioned in 5.1.

6. CONCLUSION

We presented a method for enriching community-specific procedural knowledge entries that can be found on the Web. We have achieved that by linking text fragments describing how to perform a certain action. Our experiments show that such fragments can be efficiently extracted, and that they allow for higher linking performance than state-of-the-art methods. We are releasing the dataset we used, together with our annotation, hoping that this will help in fostering further research on this subject.

Based on an analysis of the evaluation results, we also proposed a number of possible improvements, such as taking into consideration modalities during extraction and performing section-specific indexing for linking.

In addition, because of the lack of space, we did not have the opportunity to speak of the representation model that can be used to store and structure the extracted information, however releasing such a model, as linked open data, is in our immediate plans.

7. REFERENCES

[1] S. Aït-Mokhtar, J.-P. Chanod, and C. Roux. Robustness beyond shallowness: Incremental deep parsing. *Nat. Lang. Eng.*, 8(3):121–144, June 2002.

[2] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009.

[3] C. Brun, V. Nikoulina, and N. Lagos. Linguistically-adapted structural query annotation for digital libraries in the social sciences. In *Proceedings of the 6th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, LaTeCH '12, pages 55–64, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[4] Y. Gil. Human tutorial instruction in the raw. *ACM Trans. Interact. Intell. Syst.*, 5(1):2:1–2:29, Mar. 2015.

[5] B. Hachey, W. Radford, J. Nothman, M. Honnibal, and J. R. Curran. Evaluating entity linking with wikipedia. *Artificial Intelligence*, 194:130 – 150, 2013. Artificial Intelligence, Wikipedia and Semi-Structured Resources.

[6] P. Pareti, E. Klein, and A. Barker. A Semantic Web of Know-how: Linked Data for Community-centric Tasks. In *Proceedings of the 23rd International Conference on World Wide Web Companion*, pages 1011–1016, 2014.

[7] P. Pareti, B. Testu, R. Ichise, E. Klein, and A. Barker. Integrating know-how into the linked data cloud. In K. Janowicz, S. Schlobach, P. Lambrix, and E. Hyvönen, editors, *Knowledge Engineering and Knowledge Management*, volume 8876 of *Lecture Notes in Computer Science*, pages 385–396. Springer International Publishing, 2014.

[8] C. Paris, K. V. Linden, and S. Lu. Automated knowledge acquisition for instructional text generation. In *Proceedings of the 20th Annual International Conference on Computer Documentation*, SIGDOC '02, pages 142–151, New York, NY, USA, 2002. ACM.

[9] F. Roulland, S. Castellani, N. Hairon, and P. Valobra. Method and system for linking textual concepts and physical concepts, June 14 2012. US Patent App. 12/967,210.

[10] Z. Zhang, P. Webster, V. Uren, A. Varga, and F. Ciravegna. Automatically extracting procedural knowledge from instructional texts using natural language processing. In N. C. C. Chair), K. Choukri, T. Declerck, M. U. DoÄ§an, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA).