CSNIPER
Annotation-by-query for non-canonical constructions in large corpora

Richard Eckart de Castilho, Iryna Gurevych
Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science
Technische Universität Darmstadt
http://www.ukp.tu-darmstadt.de

Sabine Bartsch
English linguistics
Department of Linguistics and Literary Studies
Technische Universität Darmstadt
http://www.linglit.tu-darmstadt.de

Abstract
We present CSNIPER (Corpus Sniper), a tool that implements (i) a web-based multi-user scenario for identifying and annotating non-canonical grammatical constructions in large corpora based on linguistic queries and (ii) evaluation of annotation quality by measuring inter-rater agreement. This annotation-by-query approach efficiently harnesses expert knowledge to identify instances of linguistic phenomena that are hard to identify by means of existing automatic annotation tools.

1 Introduction
Linguistic annotation by means of automatic procedures, such as part-of-speech (POS) tagging, is a backbone of modern corpus linguistics; POS tagged corpora enhance the possibilities of corpus query. However, many linguistic phenomena are not amenable to automatic annotation and are not readily identifiable on the basis of surface features. Non-canonical constructions (NCCs), which are the use-case of the tool presented in this paper, are a case in point. NCCs, of which cleft-sentences are a well-known example, raise a number of issues that prevent their reliable automatic identification in corpora. Yet, they warrant corpus study due to the relatively low frequency of individual instances, their deviation from canonical construction patterns and frequent ambiguity. This makes them hard to distinguish from other, seemingly similar constructions. Expert knowledge is thus required to reliably identify and annotate such phenomena in sufficiently large corpora like the 100 mil. word British National Corpus (BNC Consortium, 2007). This necessitates manual annotation which is time-consuming and error-prone when carried out by individual linguists.

To overcome these issues, CSNIPER implements a web-based multi-user annotation scenario in which linguists formulate and refine queries that identify a given linguistic construction in a corpus and assess the query results to distinguish instances of the phenomenon under study (true positives) from such examples that are wrongly identified by the query (false positives). Each expert linguist thus acts as a rater rather than an annotator. The tool records assessments made by each rater. A subsequent evaluation step measures the inter-rater agreement. The actual annotation step is deferred until after this evaluation in order to achieve high annotation confidence.

CSNIPER implements an annotation-by-query approach which entails the following interlinking functionalities (see fig. 1):

**Query development**: Corpus queries can be developed and refined within the tool. Based on query results which are assessed and labeled by the user, queries can be systematically evaluated and refined for precision. This transfers some of the ideas of
relevance feedback, which is a common method of improving search results in information retrieval, to a linguistic corpus query system.

**Assessment:** Query results are presented to the user as a list of sentences with optional additional context; the user assesses and labels each sentence as representing or not representing an instance of the linguistic phenomenon under study. The tool implements a function that allows the user to comment on decisions and to temporarily mark sentences with uncertain assessments for later review.

**Evaluation:** Evaluation is a central functionality of CSNIPER serving three purposes. 1) It integrates with the query development by providing feedback to refine queries and improve query precision. 2) It provides information on sentences not labeled consistently by all users, which can be used to review the assessments. 3) It calculates the inter-rater agreement which is used in the corpus annotation step to ensure high annotation confidence.

**Corpus annotation:** By assessing and labeling query results as correct or wrong, raters provide the tool with their annotation decisions. CSNIPER annotates the corpus with those annotation decisions that exceed a certain inter-rater agreement threshold.

This annotation-by-query approach of querying, assessing, evaluating and annotating allows multiple distributed raters to incrementally improve query results and achieve high quality annotations. In this paper, we show how such an approach is well-suited for annotation tasks that require manual analysis over large corpora. The approach is generalizable to any kind of linguistic phenomena that can be located in corpora on the basis of queries and require manual assessment by multiple expert raters.

In the next two sections, we are providing a more detailed description of the use-case driving the development of CSNIPER (sect. 2) and discuss why existing tools do not provide viable solutions (sect. 3). Sect. 4 discusses CSNIPER and sect. 5 draws some conclusions and offers an outlook on the next steps.

### 2 Non-canonical grammatical constructions

The initial purpose of CSNIPER is the corpus-based study of so-called non-canonical grammatical constructions (NCC) (examples (2) - (5) below):

1. The media was now calling Reagan the frontrunner. (canonical)
2. It was Reagan whom the media was now calling the frontrunner. (it-cleft)
3. It was the media who was now calling Reagan the frontrunner. (it-cleft)
4. It was now that the media were calling Reagan the frontrunner. (it-cleft)
5. Reagan the media was not calling the frontrunner. (inversion)

NCCs are linguistic constructions that deviate in characteristic ways from the unmarked lexicogrammatical patterning and informational ordering in the sentence. This is exemplified by the constructions of sentences (2) - (5) above. While expressing the same propositional content, the order of information units available through the permissible grammatical constructions offers interesting insights into the constructional inventory of a language. It also opens up the possibility of comparing seemingly closely related languages in terms of the sets of available related constructions as well as the relations between instances of canonical and non-canonical constructions.

In linguistics, a cleft sentence is defined as a complex sentence that expresses a single proposition where the clefted element is co-referential with the following clause. E.g., *it*-clefts are comprised of the following constituents:

```
[ dummy subject it ] [ main verb to be ] [ clefted element ] [ clause ]
```

The NCCs under study pose interesting challenges both from a linguistic and a natural language processing perspective. Due to their deviation from the canonical constructions, they come in a variety of potential construction patterns as exemplified above. Non-canonical constructions can be expected to be individually rarer in any given corpus than their canonical counterparts. Their patterns of usage and their discourse functions have not yet been described exhaustively, especially not in representative corpus studies because they are notoriously hard to identify without suitable software. Their empirical distribution in corpora is thus largely unknown.

A major task in recognizing NCCs is distinguishing them from structurally similar construc-
tions with default logical and propositional content. An example of a particular difficulty from the domain of *it*-clefts are anaphoric uses of *it* as in (6) below that do not refer forward to the following clause, but are the antecedents of entities previously introduced in the context of preceding sentences. Other issues arise in cases of true relative clauses as exemplified in (7) below:

6. London will be the only capital city in Europe where rail services are expected to make a profit,' he added. *It* is a policy that could lead to economic and environmental chaos. [BNC: A9N-s400]

7. *It* is a legal manoeuvre that declined in currency in the ’80s. [BNC: B1L-s576]

Further examples of NCCs apart from the *it*-clefts addressed in this paper are *wh*-clefts and their subtypes, *all*-clefts, *there*-clefts, *if*-because-clefts and demonstrative clefts as well as inversions. All of these are as hard to identify in a corpus as *it*-clefts.

The linguistic aim of our research is a comparison of non-canonical constructions in English and German. Research on these requires very large corpora due to the relatively low frequency of the individual instances. Due to the ambiguous nature of many NCC candidates, automatically finding them in corpora is difficult. Therefore, multiple experts have to manually assess candidates in corpora.

Our approach does not aim at the exhaustive annotation of all NCCs. The major goal is to improve the understanding of the linguistic properties and usage of NCCs. Furthermore, we define a gold standard to evaluate algorithms for automatic NCC identification. In our task, the total number of NCCs in any given corpus is unknown. Thus, while we can measure the precision of queries, we cannot measure their recall. To address this, we exhaustively annotate a small part of the corpus and extrapolate the estimated number of total NCC candidates.

In summary, the requirements for a tool to support multi-user annotation of NCCs are as follows:

1. **querying** large linguistically pre-processed corpora and query refinement
2. **assessment** of sentences that are true instances of NCCs in a multi-user setting
3. **evaluation** of inter-rater agreement and query precision

In the following section, we review previous work to support linguistic annotation tasks.

### 3 Related work

We differentiate three categories of linguistic tools which all partially fulfill our requirements: *querying tools*, *annotation tools*, and *transformation tools*.

**Linguistic query tools:** Such tools allow to query a corpus using linguistic features, e.g. part-of-speech tags. Examples are ANNIS2 (Zeldes et al., 2009) and the *IMS Open Corpus Workbench* (CWB) (Christ, 1994). Both tools provide powerful query engines designed for large linguistically annotated corpora. Both are server-based tools that can be used concurrently by multiple users. However, they do not allow to assess the query results.

**Linguistic annotation tools:** Such tools allow the user to add linguistic annotations to a corpus. Examples are MMAX2 (Müller and Strube, 2006) and the *UIMA CAS Editor*¹. These tools typically display a full document for the user to annotate. As NCCs appear only occasionally in a text, such tools cannot be effectively applied to our task, as they offer no linguistic query capabilities to quickly locate potential NCCs in a large corpus.

**Linguistic transformation tools:** Such tools allow the creation of annotations using transformation rules. Examples are TextMarker (Kluegl et al., 2009) and the *UAM CorpusTool* (O’Donnell, 2008). A rule has the form `category := pattern` and creates new annotation of the type `category` on any part of a text matching `pattern`. A rule for the annotation of passive clauses in the *UAM CorpusTool* could be `passive-clause := clause + containing be% participle`. These tools do not support the assessment of the results, though. In contrast to the querying tools, transformation tools are not specifically designed to operate efficiently on large corpora. Thus, they are hardly productive for our task, which requires the analysis of large corpora.

### 4 CSNIPEr

We present CSNIPEr, an annotation tool for non-canonical constructions. Its main features are:

¹http://uima.apache.org/
Annotation-by-query – Sentences potentially containing a particular type of NCC are retrieved using a query. If the sentence contains the NCC of interest, the user manually labels it as correct and otherwise wrong. Annotations are generated based on the users’ assessments.

Distributed multi-user setting – Our web-based tool supports multiple users concurrently assessing query results. Each user can only see and edit their own assessments and has a personal query history.

Evaluation – The evaluation module provides information on assessments, number of annotated instances, query precision and inter-rater agreement.

4.1 Implementation and data
CSNIPER is implemented in Java and uses the CWB as its linguistic search engine (cf. sect. 3). Assessments are stored in a MySQL database. Currently, the British National Corpus (BNC) is used in our study. Apache UIMA and DKPro Core\(^2\) are used for linguistic pre-processing, format conversion, and to drive the indexing of the corpora. In particular, DKPro Core includes a reader for the BNC and a writer for the CWB. As the BNC does not carry lemma annotations, we add them using the DKPro TreeTagger (Schmid, 1994) module.

4.2 Query (Figure 2)
The user begins by selecting a corpus and a construction type (e.g. It-Cleft). A query can be chosen from a list of examples, from the personal query history, or a new query can be entered. The query is applied to find instances of that construction (e.g. “It” /NCC/ /PP/ /RC/). After pressing the Submit query button, the tool presents the user with a KWIC view of the query results (fig. 3). At this point, the user may choose to refine and re-run the query.

As each user may use different queries, they will typically assess different sets of query results. This can yield a set of sentences labeled by a single user only. Therefore, the tool can display those sentences for assessment that other users have assessed, but the current user has not. This allows getting labels from all users for every NCC candidate.

4.3 Assessment (Figure 3)
If the query results match the expectation, the user can switch to the assessment mode by clicking the Begin assessment button. At this point, an AnnotationCandidate record is created in the database for each sentence unless a record is already present. These records contain the offsets of the sentence in the original text, the sentence text and the construction type. In addition, an AnnotationCandidateLabel record is created for each sentence to hold the assessment to be provided by the user.

In the assessment mode, an additional Label column appears in the KWIC view. Clicking in this column cycles through the labels correct, wrong, check and nothing. When the user is uncertain, the label check can be used to mark candidates for later review. The view can be filtered for those sentences that need to be assessed, those that have been assessed, or those that have been labeled with check. A comment can be left to further describe difficult cases or to justify decisions. All changes are immediately saved to the database, so the user can stop assessing at any time and resume the process later.

The proper assessment of a sentence as an instance of a particular construction type sometimes depends on the context found in the preceding and following sentences. For this purpose, clicking on the book icon in the KWIC view displays the sentence in its larger context (fig. 4). POS tags are shown in the sentence to facilitate query refinement.

4.4 Evaluation (Figure 5)
The evaluation function provides an overview of the current assessment state (fig. 5). We support two evaluation views: by construction type and by query.

By construction type: In this view, one or more corpora, types, and users can be selected for evaluation. For these, all annotation candidates and the respective statistics are displayed. It is pos-

\(^2\)http://www.ukp.tu-darmstadt.de/research/current-projects/dkpro/
sible to filter for correct, wrong, disputed, incompletely assessed, and unassessed candidates. A candidate is disputed if it is not labeled consistently by all selected users. A candidate is incompletely assessed if at least one of the selected users labeled it and at least one other did not. Investigating disputed cases and inter-rater agreement per type using Fleiss’ Kappa (Fleiss, 1971) are the main uses of this view. The inter-rater agreement is calculated using only candidates labeled by all selected users.

By query: In this view, query precision and assessment completeness are calculated for a set of queries and users. The query precision is calculated from the labeled candidates as:

$$\text{precision} = \frac{|TP|}{|TP| + |FP|}$$

We treat a candidate as a true positive (TP) if: 1) the number of correct labels is larger than the number of wrong labels; 2) the ratio of correct labels compared to the number of raters exceeds a given threshold. Candidates are conversely treated as false positives (FPs) if the number of wrong labels is larger and the threshold is exceeded. The threshold controls the confidence of the TP and, thus, of the annotations generated from them (cf. sect. 4.5).

If a candidate is neither TP nor FP, it is unknown (UNK). When calculating precision, UNK candidates are counted as FP. The estimated precision is the precision to be expected if TP and FP are equally distributed over the set of candidates. It takes into account only the currently known TP and FP and ignores the UNK candidates. Both values are the same once all candidates have been labeled by all users.

4.5 Annotation

When the assessment process is complete, corpus annotations can be generated from the assessed candidates. Here, we employ the thresholded majority vote approach that we also use to determine the TP/FP in sect. 4.4. Annotations for the respective NCC type are added directly to the corpus. The augmented corpus can be used in further exploratory work. Alternatively, a file with all assessed candidates can be generated to serve as training data for identification methods based on machine learning.

5 Conclusions

We have presented CSNIPER, a tool for the annotation of linguistic phenomena whose investigation requires the analysis of large corpora due to a relatively low frequency of instances and whose identification requires expert knowledge to distinguish them from other similar constructions. Our tool integrates the complete functionality needed for the annotation-by-query workflow. It provides distributed multi-user annotation and evaluation. The feedback provided by the integrated evaluation module can be used to systematically refine queries and improve assessments. Finally, high-confidence annotations can be generated from the assessments.
The annotation-by-query approach can be generalized beyond non-canonical constructions to other linguistic phenomena with similar properties. An example could be metaphors, which typically also appear with comparatively low frequency and require expert knowledge to be annotated. We plan to integrate further automatic annotations and query possibilities to support such further use-cases.

Acknowledgments

We would like to thank Erik-Lân Do Dinh, who assisted in implementing CSNIPER as well as Gert Weibelhuth and Janina Rado for testing and providing valuable feedback.

This work has been supported by the Hessian research excellence program “Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz” (LOEWE) as part of the research center “Digital Humanities” and by the Volkswagen Foundation as part of the Lichtenberg-Professorship Program under grant No. I/82806.

Data cited herein have been extracted from the British National Corpus, distributed by Oxford University Computing Services on behalf of the BNC Consortium. All rights in the texts cited are reserved.

References