Honing the Sketch Engine Prepositions

Ken Litkowski
CL Research
9208 Gue Road
Damascus, MD 20872 USA
ken@clres.com

Abstract

The initial construction of the vertical file for Sketch Engine (SE) for the corpus of the Pattern Dictionary of English Prepositions (PDEP) contained 99 percent of the instances. We have refined the original Python script used to create the SE files to understand the difficulties with the missing instances and make further examination of problematic identification of the complements and governors in other instances. After making these refinements, we have updated the SE vertical file. In addition, we have incorporated these refinements in the PDEP improvements in the files that can be downloaded to interested parties. The original downloaded data consisted of the MySQL files for the PDEP parses and feature files associated to the preposition instances. Here, we now make available the updated MySQL files, the refined Python script for creating the vertical file, the outputted messages and problems generated during the script, and a help file describing each corpus and detailing the hierarchical script functions. After describing the script refinements, we present plans for PDEP: improvements in PDEP supersenses, reviewing the corpora tagging, completing further fields in the preposition patterns, analyzing substitutable prepositions using preposition digraphs, and extending preposition idioms in multiword expressions.

1. Introduction

Litkowski (2017) describes the Pattern Dictionary of English Prepositions (PDEP, Litkowski (2014)) and the procedures used to create the files to be installed into Sketch Engine[1] (SE). Litkowski (2017) also describes how to use SE for the preposition corpus. That paper also describes, in its appendix, the Python script and all its subroutines. In section 2, we describe the changes in the Python script, particularly how the changes report on each instance and provide the basis for examining the results. In section 3, we summarize and characterize each corpus, providing an overview of the PDEP instances. Section 4 describes the new downloadable data for PDEP: descriptions of the generated data, details of the MySQL files, and descriptions of the corpora and the functions and subroutines. Based on the overview, in section 5, we describe plans for further developments for PDEP with the general objection of providing an overall characterization of prepositions.

2. Changes in the SE Script to Create Vertical Files

In the initial implementation of the **download_parses.py**, some information about the instances was printed to standard error (**stderr**) or standard output (**stdout**). This information was not saved, but this

[1] https://www.sketchengine.eu/

output provided an opportunity for a more detailed examination of the processes. We modified the script to capture this data in message and problem files to provide further investigation.[2]

The main function for the script was to create vertical files for the parses for each instance available at PDEP. These parses, using the dependency CoNLL-X format, did not include certain basic data about each instance, specifically, the preposition name, the corpus, the preposition sense, and the corpus instance number. We also add information about the class and the subclass for the preposition sense, the lemma of the complement and the governor, and a supersense tag for the complement and the governor. In the remaining paragraphs in this section, we describe the changes in each of functions and subroutines.

The **main** function gets the complement and the governor data and the parses for each preposition. There is no substantive difference in this function, except printing to a message file identification for the preposition, using the same initial line for the preposition, i.e., the XML **doc** element with the corpus name and the preposition file name. This function contains the primary function, **get_content**, which processes the parse file for each preposition, describing further changes in the next paragraph.

The essence of **get_content** is chiefly the same, first getting the JSON complement and governor data for each instance (printing this data in a message file), then getting each sentence (involving linking it to its parse), and finally processing the file for the tokenized sentence into the vertical file (with the function **create_new_vert**). This function has three minor differences. (1) Some instances are unable to find the complement or the governor. In the original, if the governor could not be found, no attempt was made to search for the complement; the modification tried for each, rather than in one simultaneous attempt. This modification obtained some instances for which a complement could be found. (2) Part of the process for linking the sentence instance with the complement-governor instance made a call of **prepsents.php** for each sense. This script could call for "all" senses than each one (e.g., one call rather than 250 calls). This modification was simply a little bit more efficient. (3) In some cases, it was not possible to find a matching sentence for the specified sense. This fact was part of **stderr**, but otherwise not recorded. This fact was also printed out for the problem file, for use for further examination of the reason.

The function **create_new_vert** continued the procedure for adding each token using the parse, i.e., the token identification, the token word, the lemma (with a single letter part of speech), the part of speech, the dependency identification, and the syntactic type. Before the tokens were entered into the vertical file, the function **include_component** was the first step in this function to identify the starting and the ending locations of the preposition, the complement, and the governor. The function was modified in two ways: (1) adding XML elements and their attributes to the vertical file and (2) recording problems in not being able to close elements for an instance. A **<s>** (sentence) element was entered before any tokens were added to the vertical file for a sentence. Attributes were added to the **<s>** element for the preposition sense (**sense_label**), the class of the sense (**class**), the subclass (**subc**) if available, the instance number (**inst**), and a link to allow a user to examine the PDEP pattern fields.[3] The function continued putting opening and ending element names before and after the token in the vertical file (**<prep>**, **<compl>**, and **<gov>**). In the modification, a supersense tag attribute (**sst**) was added to the complement and the governor tokens when one could be identified. The method of assigning a supersense used the WordNet

[2] The initial script was developed by Marek Medved at Lexical Computing, Ltd. This effort provided the basis for making it easy to make several refinements that improved PDEP data.
[3] E.g., https://clres.com/db/TPPpattern.php?prep=abaft&sense=1(1)

lexicographer file name, as in McCarthy et al., (2015). The second change of this function recorded any problem in finding the ending element of the preposition (**</prep>**), the complement (**</compl>)**, or the governor (**</gov>**).

The function **include_component** tries to find the starting and the ending token for the preposition, the complement, and the governor. To accomplish this, the important arguments are the "plain sentence" (a string removing the spaces of the sentence) and the desired structure. In some cases, an empty "plain sentence" was the value of the argument. In other cases, it was not possible to begin and/or end the desired structure (i.e., the preposition, the complement, or the governor). In these cases, the function returned the value "-1" for the start and the end as the result of the function. There was no modification of this functionality, but rather we wrote a line to the problem file, indicating the corpus, the preposition, the sense, the instance, and the structure that couldn't be found.

No other functions were modified. In summary, only a few minor changes were made to the script content. The primary changes were printing out to identify the instances that could not be fully characterized. By doing this, we were able to allow us to summarize statistics about the effectiveness of the script and to identify the problematic instances.

3. Characterizing Instances in the PDEP Vertical Files

The script to create the PDEP vertical file for SE involved three runs, once for each subcorpus: the Corpus Pattern Analysis (CPA/TPP), the Oxford English Corpus (OEC), and the FrameNet Corpus (FN). Details for each corpus and the script are shown in the web pages for "Preposition Corpora in Sketch Engine."[4] The page "Corpora Sources" provides the overall number of the initial instances and the number of instances that were actual included in PDEP. The links in this page provide the details for each of the three corpora. The first web page under each corpus identifies the total number of instances in the corpus and identifies the number of instances that were not included in PDEP. The omitted instances and the reasons for the omissions are described in the other web pages. The changes in the script made it possible to understand the omissions and the improvements in the vertical files.

Each corpus has a web page that describes statistics for the vertical file created by the script. The current set of the runs is now the fourth version and is compared with the third version. The datum is the length of running the **time** for each corpus; the total for all the corpora is about 50 minutes. We list the number of **lines** for each corpus; increases in the number of lines generally correspond to the ability to identify more elements for complements or governors. Generally, the number of **tokens** was about the same between versions; we improve the tokenization for the OEC and the FN corpora, accounting for the larger number of tokens in these two corpora. The numbers of **sentences** and **prepositions** were generally the same within each version. The numbers of sentences and prepositions are the same between versions for the CPA/TPP and the OEC corpora, but there are 300 fewer in the earlier version of the FN corpus.

The **complements** were found for approximately 97 percent (78114/80695) of the instances for the three corpora. **Complements with supersenses** were tagged for 85 percent (66145/78114) for the three corpora. The main reasons for the fewer instances with supersenses were the occurrence of pronouns and proper nouns (i.e., not having semantic WordNet senses). **Governors** were found for approximately the

[4] https://www.clres.com/db/ske/CorpusAnalysis.html

same as the percentages of the complements, about 96 percent (77792/80695). The percentages of **governors with supersenses** were many fewer instances, again 84 percent (64977/77792), because of pronouns and copular verbs.

As mentioned above in discussing the function **get_content**, in some cases, it was not possible to find a matching sentence for the specified sense, and printed a problem beginning with "Not found for record". We indicated that this would be examined further. These instances are different for the three corpora. For CPA, the full set of the 56 instances for the preposition *vis-à-vis*, for which the features were not generated, possibly because of the hyphen in the name of the preposition. For OEC, there are 27 instances not included in the vertical file, 11 because of a double quote mark which caused a problem in constructing the "plain sentence" with the Python treatment and 16 because the features were not created for the sentences. In FN, there are 197 instances in the FN corpus. 41 because of faulty sentences and 156 because the preposition had not been tagged.

The problem files also identify several other types. There were 4461 occurrences of the type "**Can't include structure**" (when it was unable to create a plain sentence and thus did not have the structure) for complements (1600) or governors (2861). There were 3939 CPA instances (1266 for the complements and 2613 for the governors), 229 OEC instances (115 for the complements and 114 for the governors), and 293 FN instances (219 for the complements and 74 for the governors). There were 1029 occurrences of the type "**Can't include component**" (when some intervening characters, such as a pound sign, made it impossible to find the component) for the preposition (6), the complement (981), or the governor (42). There were 1013 CPA instances (3 for the prepositions, 968 for the complements, and 42 for the governors), one OEC instance (one preposition), and 15 FN instances (2 prepositions and 13 complements). There were 69 occurrences of the type "**Can't end structure**" (when the starting position for a component was identified, but the last token did not have an ending element) for the preposition (19), the complement (15), or the governor (35). There were 63 CPA instances (15 for the preposition, 13 for the complement, and 35 for the governor); no such OEC instances; and 6 FN instances (4 for the prepositions and 2 for the complements).

As indicated above, we print messages identifying the character location and the length of the complement and the governor for each instance. There are 80975 such messages out of 81509 instances (534 instances were degenerate). One or both of the complement and the governor could not be identified for 4224 (5.2%) instances: 322 FN instances, 240 OEC instances, 3662 CPA instances. Both were missing for 457 instances. It should be emphasized that these location and length messages are not necessarily accurate for the complements and the governors. And there may be other inaccuracies. Some instances have the location and length for both the complement and the governor, indicating that there are some problems with the dependency generations.

4. New Available Downloadable PDEP Data

The initial downloadable PDEP data[5] consisted of the three MySQL database tables for the sense inventory, the PDEP properties for each sense, and the tagged corpora containing all instances. This file also included three papers describing the original TPP corpora (Litkowski, 2013), the PDEP paper

[5] https://www.clres.com/elec_dictionaries.html#pdep. Note that this link now updates with the MySQL database tables. This reference also includes an additional link to download the material described in this paper.

(Litkowski, 2014), and a paper (Schneider et al., 2015) describing the initial creation of preposition supersenses, based on creating the PDEP supersense field. A README.txt file provided more details about the contents of the zip file.

With this paper, the downloadable file updates the MySQL tables and includes additional information, focusing on the vertical file used to create the English preposition corpus in Sketch Engine. All the data described below are compressed into a WinZip file.

- Creating the English preposition corpus
  - The initial vertical file was created from the original Python script[6]. This script was reimplemented locally[7].
    - The basis for the script used html files in a **data** folder. There was one file for each subcorpus (**cpa.html**, **fn.html**, and **oec.html**) identifying each preposition.
    - The data used for creating supersenses for complements and governors (from WordNet) were obtained from GAZ files[8] (described in Ciarimita and Altun (2006)).
    - The vertical file was created in the **output** folder with four **vert** files, three for each of the subcorpora and one containing the full subcorpus.
    - The **messages** folder contains four **msgs.txt** file, three for each of the subcorpora and one containing the full subcorpus.
    - The **problems** folder contains four **probs.txt** file, three for each of the subcorpora and one containing the full subcorpus.
- The MySQL files consist of three database tables:
  - The **Preposition Sense Inventory** table (prepdefs.sql) consists of three fields: the preposition, a sense number, and the definition. The total number of senses in prepdefs.sql is 1040.
  - The **Preposition Properties** table (prepprops.sql) consists of 27 fields used to describe each sense (preposition pattern). An overview of these fields is included in the PDEP help file (https://www.clres.com/pdep.html#patterns). A more detailed description is given when a pattern is displayed, where hovering the mouse will provide a description of each field. The original set of fields in TPP may be examined at https://www.clres.com/cgi-bin/onlineTPP/find_prep.cgi.
  - The **Preposition Tagged Corpora** table (prepcorp.sql) consists of six fields: the preposition, the source (FN, OEC, CPA), the sense tag, the instance number, the 0-based location where the preposition begins, and the sentence text. There are 82,329 sentences, with 7485 in OEC, 26739 in FN, and 48105 in CPA.
- Detailed descriptions about the PDEP Sketch Engine in Web Help contains two topics
  - Corpora Sources: Describes each of the three corpora in details, with a general description, details of the instances that were not included in the vertical file, and details about the processing of the vertical file for each corpus.
  - Prepositions in Sketch Engine: Describes the functions of the Python script (as a modification of the original script) used to create the vertical files. The description shows the hierarchy of the various functions.

[6] download_parses.py as described above, from Lexical Computing. Ltd.
[7] skevert1.py
[8] adjs.gaz, advs.gaz, nouns.gaz, and verbs.gaz

5. Future PDEP Plans

The usefulness of the PDEP data can be improved. Several areas of research can be identified. Any comments, suggestions, or help about the areas described below would be appreciated.

- Further analysis of supersenses: Schneider at al. (2018) describes the benefit of supersense disambiguation of prepositions, with detailed guidelines for using the 50 supersenses (Schneider et al., 2017). As described above, PDEP contains the earlier version of the supersenses, but have not yet been completely mapped to the current set. Litkowski (2018b) describes this process[9], but also raises questions of the consistency of the new supersenses. The consistency can be further examined by looking at the tagging between the first and second versions. Part of the importance of the consistency analysis not only will help identifying the new supersenses in PDEP, but also may ascertain the extent to which the newer supersenses will benefit an analysis of the tagging in the PDEP corpora.
- Checking the tags for the CPA/TPP corpus: Tagging the CPA/TPP corpus was performed by one person. The other two corpora were performed by professional lexicographers (the OEC by staff of Oxford University Press and FN by Orin Hargraves). The tags for these corpora were frequently used in tagging the CPA/TPP instances. As a result, the professional tags can be viewed with higher confidence. It would be useful to assess the confidence of the CPA/TPP tagging, particularly for the polysemous prepositions. Several prepositions have only a single sense, where we can assign a higher confidence. In addition, close attention to the sense hierarchies within the Oxford senses can be further examined to change the confidence. For example, the sense *against (7(2c))* requires a complement that is "an amount of money"; in such a case, we would assess instances with such tags as having higher confidence.
- More completion of pattern properties: Many fields for the sense patterns have not yet been completed. These fields need information for characterizing complement and governor selectors (assuring that a sense selection is correct), syntactic characteristics of complement and governor. With the creation of WordNet supersenses via the creation of the vertical files for about 85 percent of the complements and governors, these fields can be filled that will provide additional information in patterns.
- Analysis of substitutable prepositions: In the TPP development, the lexicographer identified other prepositions that have similar senses (i.e., components of meaning for the senses). In addition, the lexicographer identified "Generic Classes of Prepositions"[10]. (These classes have been refined and adjusted during PDEP in "Preposition Classes Analyses"[11].) As part of the 20 generic classes, the name of each class was linked in a directed graph (digraph) showing how the preposition sense in that class are related, from primitive senses to those that are derived from the primitives. This work is based on the early TPP developments (Litkowski, 2002; Litkowski and Hargraves, 2005)
- Preposition idioms: In discussing Schneider et al. (2017), we observed that the developers were observing and tagging some idiomatic phrases (e.g., *by far* and *for free*) with their new

[9] See also in https://githum.com/kenclr/pdep2psst2
[10] https://www.clres.com/PrepositionClasses.htm
[11] https://www.clres.com/db/classes/ClassAnalysis.php

supersenses. We suggested that looking up the definitions for such phrases in the *Oxford Dictionary of English* (ODE) might help to identify the appropriate supersense. This led us to examine the 120,000 dictionary entries in ODE and observed that almost 5300 multiword expressions (MWEs) might constitute preposition-phrase (PP) idioms that either begin or end prepositions. Many, possibly most, of these idioms may have some relationship to the PDEP sense inventories for individual prepositions. We have discussed this initial analysis of such phrases (Litkowski, 2018a). We are planning to discuss steps with OUP to allow a more detailed examination of the issues involved in that topic.

References

Massimiliano Ciaramita and Yasemin Altun. 2006. Broad-Coverage Sense Disambiguation and Information Extraction with a Supersense Sequence Tagger. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Sydney, Australia, 594-602 (http://www.aclweb.org/anthology/W-6-1670)

Ken Litkowski. 2018a. *The Analysis of PP Idioms*. Technical Report 18-01. Damascus, MD: CL Research. (https://www.clres.com/online-papers/PPIdioms.pdf) (See also support data at GitHub Data.)

Ken Litkowski. 2018b. *Updating Supersenses in the Preposition Pattern Dictionary*. Technical Report 18-02. Damascus, MD: CL Research. (https://www.clres.com/online-papers/PSST2.pdf) (See also supporting data at GitHub Data.)

Ken Litkowski. 2017. *The Preposition Corpus in Sketch Engine*. Technical Report 17-01. Damascus, MD: CL Research. (https://www.clres.com/online-papers/PrepsSkE.pdf)

Ken Litkowski. 2014. Pattern Dictionary of English Prepositions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Baltimore, Maryland, USA, pp. 1274-83. (https://aclweb.org/anthology/P14-1120)

Ken Litkowski. 2013. *The Preposition Project Corpora*. Technical Report 13-01. Damascus, MD: CL Research. (https://www.clres.com/online-papers/TPPCorpora.pdf)

Kenneth C. Litkowski. 2002. Digraph Analysis of Dictionary Preposition Definitions. In *Proceedings of the ACL-02 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions.* Philadelphia, PA, pp. 9-16. (https://aclweb.org/anthology/W02-0802)

Kenneth C. Litkowski and Orin Hargraves. 2005. The Preposition Project. In *Proceedings of the Second ACL-SIGSEM Workshop on The Linguistic Dimensions of Prepositions and their Use in Computational Linguistics Formalisms and Applications.* Colchester, England: University of Essex. (https://www.clres.com/online-papers/sigsem2005.pdf)

Diana McCarthy, Adam Kilgarriff, Milos Jakubicek, and Siva Reddy. 2015. Semantic Word Sketches. *Corpus Linguistics 2015*. Lancaster University.

Nathan Schneider, Jena Hwang, Vivek Srikumar, Jakob Prange, Austin Blodgett, Sarah H. Moeller, Aviram Stern, Adi Bitan, and Omri Abend. 2018. Comprehensive Supersense Disambiguation of English Prepositions and Possessives. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia, pp. 185-196. (https://aclweb.org/anthology/P18-1018)

Nathan Schneider, Jena Hwang, Archna Bhatia, Na-Rae Han, Vivek Srikumar, Tim O'Gorman, Sarah H. Moeller, , Omri Abend, Austin Blodgett, and Jakob Prange. 2017. Adposition and Case Supersenses v2: Guidelines for English. arXiv:1704.02134.

Nathan Schneider, Vivek Srikumar, Jena Hwang, and Martha Palmer. 2015. A Hierarchy with, of, and for Preposition Supersenses. In *Proceedings of the 9th Linguistic Annotation Workshop*. Denver, Colorado, pp. 112-123.