Enrichment of Inflection Dictionaries: Automatic Extraction of Semantic Labels from Encyclopedic Definitions

Pawel Chrzaszcz

Computational Linguistics Department, Jagiellonian University, Golebia 24, Kraków, Poland
Computer Science Department, AGH University of Science and Technology,
Mickiewicza 30, Kraków, Poland

Abstract. Inflection dictionaries are widely used in many natural language processing tasks, especially for inflecting languages. However, they lack semantic information, which could increase the accuracy of such processing. This paper describes a method to extract semantic labels from encyclopedic entries. Adding such labels to an inflection dictionary could eliminate the need of using ontologies and similar complex semantic structures for many typical tasks. A semantic label is either a single word or a sequence of words that describes the meaning of a headword, hence it is similar to a semantic category. However, no taxonomy of such categories is known prior to the extraction. Encyclopedic articles consist of headwords and their definitions, so the definitions are used as sources for semantic labels. The described algorithm has been implemented for extracting data from the Polish Wikipedia. It is based on definition structure analysis, heuristic methods and word form recognition and processing with use of the Polish Inflection Dictionary. This paper contains a description of the method and test results as well as discussion on possible further development.

1 Introduction

Typical natural language processing (NLP) algorithms rely on word frequency statistics (e.g. TF-IDF). More advanced processing requires the use of feature extraction. For inflecting languages, like Polish, the basic features are: parts of speech, headwords and grammatical categories such as gender, case etc. They are used to tag the document text. Statistical algorithms may use tags to process the information encoded in the text, e.g. for authorship resolution [1]. Tags in the text may be also used to train the algorithm, which will automatically tag the rough untagged text. For example the SVM classifier trained on a large corpus may be used to tag Polish text with parts of speech with 96-97% accuracy [6] and ensemble methods increase that level, so the tags can be expanded to grammatical categories [7].

To eliminate the inherent significant error rate of statistical methods, one may use a morphological analyzer which may also allow to extend the tag structure, introducing for example gender, case etc. – popular examples of such tools for Polish are Morfeusz

Chrzaszcz P..
Enrichment of Inflection Dictionaries: Automatic Extraction of Semantic Labels from Encyclopedic Definitions.
DOI: 10.5220/0004100501060119
In *Proceedings of the 9th International Workshop on Natural Language Processing and Cognitive Science* (NLPCS-2012), pages 106-119
ISBN: 978-989-8565-16-7

[16] and Morfologik[1]. As an alternative for such taggers one may use the Polish Inflection Dictionary (SFJP)[2]. One of the possible forms of a dictionary is a database, which means that one can automatically expand information stored in the dictionary introducing semantic information. Figure 1 shows an example of use of such an expanded dictionary. The example sentence is “Wyszedlem z psem na spacer” (I went for a walk with my dog). The first block shows features extracted for the word form “psem” using only an inflection dictionary: the lemma (“pies” – a dog) and all grammatical categories are extracted. This information could be sufficient in some cases, e.g. to decide that the sentence contains information about dogs.

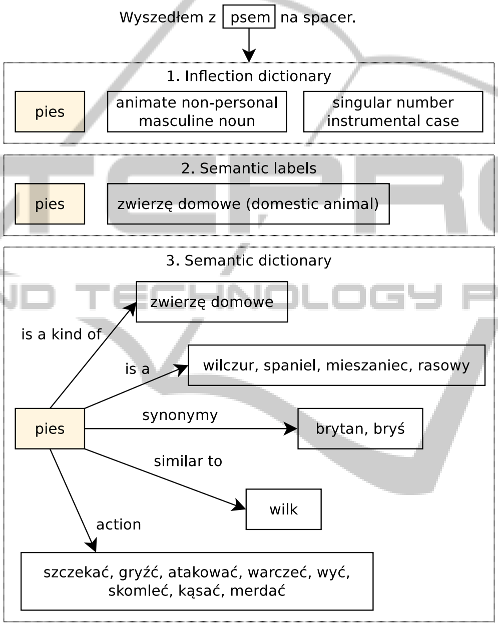
Wyszedłem z psem na spacer.
1. Inflection dictionary
pies
animate non-personal masculine noun
singular number instrumental case
2. Semantic labels
pies
zwierzę domowe (domestic animal)
3. Semantic dictionary
zwierzę domowe
is a kind of
is a
wilczur, spaniel, mieszaniec, rasowy
pies
synonymy
brytan, bryś
similar to
wilk
action
szczekać, gryźć, atakować, warczeć, wyć, skomleć, kąsać, merdać

Fig. 1. Feature extraction for the word “psem” using an inflection dictionary, semantic labels and a semantic dictionary.

However, syntactical text processing is often not enough. To recognize semantic relations between words, one needs a source of semantic information. Examples of such resources are ontologies, e.g. CYC[3], but they often focus on a complicated taxonomy of entities while not containing syntagmatic relations or connections between nouns and verbs. Moreover, it takes a lot of time to construct an ontology, the result is always incomplete and there are difficulties with connecting the ontology to an inflection dictionary. There are also dictionaries like WordNet [3], for which there is a Polish

[1] http://morfologik.blogspot.com

[2] SFJP is a dictionary of Polish language developed by the Computer Linguistics Group at AGH University of Science and Technology in Kraków, in cooperation with the Department of Computational Linguistics at the Jagiellonian University [8]. It contains more than 120 thousand headwords and provides a programming interface – the CLP library [4].

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

version[4], but this dictionary also lacks syntagmatic relations [12]. Finally, there is an ongoing process of creating the Polish Semantic Dictionary (SSJP), connected with SFJP. However, it will probably also never be as complete as inflection dictionaries. The third block in Figure 1 shows semantic relations for the word "pies" (from SSJP). As we can see, SSJP introduces syntagmatic relations, which connects for example *a dog* and *barking*. That makes it possible to say that the sentence: "Azor szczekal, wiec szybko wrócilem ze spaceru" (*Azor was barking so I quickly returned from the walk*) is also about a dog. It is clear that there is rich information that allows much more accurate information retrieval than with the inflection dictionary alone, but creation of such a resource requires a lot of tedious manual work.

It is much easier to introduce some semantic information into an inflection dictionary. For example, it is possible to automatically extract semantic labels describing meaning of words from an encyclopedia or a dictionary. The labels would become the middle layer in the hierarchy shown in Figure 1. Although this semantic information is not rich, it should be a significant improvement over simple syntactical processing. For instance, almost all breeds of dogs appear in an encyclopedia, so if the label "pies" is extracted for all of them, it would be possible to recognize all dog breed names in the source text. Other examples are proper names (towns, mountains, companies etc.) that are often missing from ontologies.

The goal of this paper is to show a method of automatic extraction of such semantic labels from the Polish Wikipedia. Firstly, we present the motivation for choosing this particular resource as the source of semantic information. Next, the extraction algorithm is described. To assess the efficiency of that method, several tests were performed. We provide their results and discuss them. Finally, we describe how the resulting data can further be processed and how much room for improvement there is left.

1.1 Wikipedia as a Source of Semantic Information

The simplest resource of any linguistic data is plain text – one can use a corpus for extracting semantic information. There are a few such corpora for Polish language, containing millions of words. However, extracting any semantic labels or categories from unstructured text is a difficult task [11]. It is much better to use a resource that already contains word definitions – an encyclopedia. A decision was made to choose the Polish Wikipedia[5] as the data source. The main reasons for this choice are: openness, maturity and size of this online resource.

The Polish Wikipedia contains already more than 800 thousand articles, which is much more than, for example, in Wielka Encyklopedia PWN, which consists of 30 printed volumes and contains about 140 thousand headwords. The number of entries in Wikipedia has been increasing at a constant rate since 2007 (about 100 thousand new articles are added each year) – this proves that it is a stable and mature project. Openness of Wikipedia enables anyone to edit it, so the articles are changing quickly according to the latest events and news. On the other hand, there is a risk of vandalism and low quality of entries – some of them contain multiple language, structural and

[4] http://plwordnet.pwr.wroc.pl

[5] http://pl.wikipedia.org

factual errors. However, there are some means of controlling the quality of editions and they are apparently becoming better over time[6].

The use of Wikipedia as a linguistic resource is becoming more and more common, slowly replacing other semantic data sources in certain applications. For example, using WordNet for expression disambiguation often yields average results because of imperfect disambiguation methods [15] and fine granularity of WordNet classification [2]. Using Wikipedia instead of WordNet may significantly rise the accuracy of such disambiguation [10]. Wikipedia can also be used as a source for creating ontologies – examples include YAGO [13] and DBPedia[7]. Article content, infoboxes, page categories and relations between pages (various types of links) are commonly used as the source data [9]. First sentences of articles are also often used as a source of word definitions. For example, J. Kazama and K. Torisawa [5] analyzed definitions from the English Wikipedia to disambiguate proper names. A. Toral and R. Muñoz [14] used the Simple English Wikipedia[8] to categorize proper names and match the category names with WordNet entries – the topic of that work is closely related to this paper. However, the simple algorithm used for English is not suitable for an inflecting language, which caused the need of designing a new one for Polish.

2 Label Extraction

The basic unit of the output of the extraction algorithm is a **semantic label** – a short definition consisting of a single word or a sequence of words. It should be as short as possible while retaining the meaning of the definition. For example, a good semantic label for “Kraków” is “miasto” (*a city*) and for “Lance Armstrong” – “kolarz szosowy” (*a road cyclist*). The main part of the label is the head noun. If a single noun is not enough to provide the full definition, additional adjectives and nouns may be added. For example, the meaning of the label “pilkarz reczny” (*a handball player*) is completely different from the head noun “pilkarz” (*a footballer*). For some words, it is easier to provide an indirect definition that uses some additional relations, e.g. “grupa ludzi” (*a group of people*), “rasa kota” (*a breed of cat*), “czesc samochodu” (*a part of a car*) – in these cases the operators of these relations should be included in the label.

The input data is the Wikipedia content, which consists of individual articles. A typical Wikipedia article starts with a title (a headword), followed by a description. The first paragraph of this description usually begins with a short definition of the headword, which is used as the source of semantic labels. There are two special kinds of pages in Wikipedia.

1. **Disambiguation Pages.** One headword may have several meanings (homonymy). To disambiguate them, an additional disambiguation phrase in parentheses is added. An example is the polish word “kreda” which can mean either *Cretaceous* or *chalk*. The article about *Cretaceous* is titled “Kreda (okres)” (*a period*) and the latter is called “Kreda (skala)” (*rock*). To provide access to these pages, the headword

[6] System wersji przejrzanych, http://pl.wikipedia.org/wiki/Wikipedia:Wersje_przejrzane

[7] http://wiki.dbpedia.org

[8] http://simple.wikipedia.org

“Kreda” leads to an additional disambiguation page which contains links to all possible meanings. During the processing the disambiguation page is skipped and the data from the remaining pages are saved with identical headwords, but annotated with the corresponding disambiguation phrases. Sometimes one of the meanings is the most common, for example “kot” (*a cat*) is generally an animal, but there also exists a small lake with the same name. In this case the disambiguation page is called “Kot (ujednoznacznienie)” (*disambiguation*) and the article about an animal is called simply “Kot”, so the data for this page is saved with an empty disambiguation phrase, which means that this is the primary meaning of the headword “Kot”. The article about the lake is titled “Kot (jezioro)” (*a lake*).

2. **Redirections.** If several headwords have the same meaning, all lead to the same article. However, one of them is a direct link and others are redirections to that one (e.g. “Buk pospolity” redirects to “Buk zwyczajny”). This redirection graph should be saved, because it can be used to find groups of headwords with the same meaning[9]. However, in some cases these links will need to be broken, because sometimes the meanings of the source and the destination may differ.

2.1 Extracting the Headword and its Description

Both the headword and the first article paragraph can be broken into individual *tokens* and stored as a *token list*. Tokens are words, numbers or punctuation marks. Parentheses are a special kind of characters – they not only divide the text into smaller fragments, but also create an independent text part enriching the main text with some extra information. The text is still meaningful without these bracketed fragments. To allow text processing on different levels of detail, all bracketed fragments are stored as sublists in the main token list, resulting in a data structure called a *token tree*. Wikipedia contents contain frequent errors, which include missing opening or closing parentheses, so the token tree construction algorithm has to skip that redundant unbalanced brackets.

The first paragraph of a Wikipedia article starts with a repeated headword – this is a common convention used in encyclopedic articles. The rest of the paragraph is usually a description of the object, which is the place where the definition can be found. Unfortunately, exceptions from this structure are not rare: the beginning of the first paragraph often differs from the headword. To solve this issue, a special algorithm was developed. It tries to match the headword on four levels. If matching on a certain level succeeds, the algorithm breaks and returns the *offset* – index of the first token after the matched headword at the top level of the paragraph token tree.

Level 1. *Full Match.* the headword matches exactly the beginning of the paragraph. An example is shown below. The vertical line indicates the offset (in all examples).

```
FIS Team Tour
FIS Team Tour |(znany równiez jako druzynowy Turniej Tr
zech Skoczni) - zawody w skokach narciarskich (...)
```

Level 2. *All Words.* Most of the headwords are multipart words. The order of the tokens can sometimes be changed without losing the original meaning. It means that the

[9] It is worth noting that both the source and the destination may contain disambiguation phrases.

author can, either intentionally or not, put headword components in a changed order at the beginning of the paragraph. Sometimes also new words will be added, some words will be in parentheses and punctuation marks will differ. This is why on the second match level the headword token list is converted to a set of words and a search for these words is performed in the paragraph token tree. If all words are found, the match is successful. The distance between matched words, measured at the top level of the paragraph token tree, must not be greater than 2[10]. Without that condition some random matches would occur, like in the following example:

```
Zamek Ksiazat Pomorskich w Ueckermünde
Zamek Ksiazat Pomorskich (niem. Schloß der Herzöge von
Pommern) - ostatni z zachowanych zamków ksiazat pomorsk
ich na obszarze obecnych Niemiec znajduje sie w Uecker
münde |na Pomorzu Przednim.
```

The article is about a castle in Ueckermünde. The phrase "w Ueckermünde" is omitted in the paragraph, because it is an optional part of the name. However, it was accidentally found at the end of the paragraph, where it was used to describe the location of this building. As a result, the offset is too high and the part containing the definition "ostatni z zachowanych zamków" (*the last one of preserved castles*) is skipped. After introducing the distance limit, only the first three words will be found (the distance to the next one is 12) and the match on the second level will fail.

Level 3. *Acronyms.* Sometimes the title contains acronyms which are expanded in the paragraph. Matching on this level is similar to the previous one, but each word consisting only of capital letters is treated as an acronym and the search is performed for both the original word and a sequence of words starting with the capital letters from the acronym.

Level 4. *Similar Words.* On this level the algorithm is similar to the previous one, but two words match not only if they are identical, but also if the Levenshtein distance between them is lower than a threshold value (20% of the length of the word from the headword). Another difference is that this last matching succeeds if any of the words is matched. It turns out that this approach results in higher accuracy than a more strict condition, because it is better to skip a partial name than to search for the definition in it.

If the offset is greater than zero but the definition cannot be found, another search is performed for zero offset. This allows searching for the definition in the headword, which is reasonable for some self-describing titles where no typical definition is included in the paragraph. These exceptions include symbols (flags, crests) and public institutions (schools, churches). For example, the following article is about the flag of the town of Ostroleka. It describes the pattern of the flag instead of explaining what a flag is, so no definition can be found after the offset. However, the word "flaga" (*a flag*) is an acceptable definition in this case.

```
Flaga Ostroleki
Flaga Ostroleki |sklada sie z trzech poziomych, równole
glych pasów, o równej szerokosci i dlugosci.
```

[10] Tests with different values have been performed and this value resulted in best performance.

2.2 Dividing the Description into Sentences and Fragments

The part of the paragraph starting at the offset is supposed to contain the definition. That definition is most likely to be found at the top level of the paragraph token tree, so all the deeper levels can be ignored. Resulting token list is then broken into sentences[11]. The first sentence is then used to search for the definition. Remaining sentences are ignored, because the probability that they will contain the definition is very low (tests with 2 and 3 first sentences yielded worse results).

The starting point of the search should not be always at the beginning of the sentence. For example, in the following example the label "zaburzenie osobowosci" (*personality disorder*) appears after a synonym of the headword – "osobowosc obsesyjno-kompulsywna".

```
Osobowosc anankastyczna
Osobowosc anankastyczna|, osobowosc obsesyjno-kompulsyw
na - zaburzenie osobowosci, w którym wystepuje wzorzec
zachowan zdominowany dbaloscia o porzadek, perfekcjoniz
mem (...)
```

The part of the sentence that may contain the definition is called a *sentence fragment*. Multiple tests and analysis resulted in the following heuristics. There are special tokens that were found to appear at the beginning of fragments:

1. Punctuation marks: — (*em dash*), – (*en dash*), - (*hyphen*), ":" (*colon*), "," (*comma*), "." (*full stop*), each of them has to be followed by a white space.
2. The word "byc" (*to be*) in the present or past tense: *jest, sa, byl, bylo, byla, byly*.
3. The word "to" (*it* or *this*).

Each fragment starts with zero or one character from the first group, followed by zero or one word from the second group, followed by zero or one word from the third group. This sequence will be called a *fragment prefix*. A set of all possible non-empty disjoint prefixes induces a set of all possible fragments. That set needs to be sorted according to descending probability of containing the definition. Best results were observed for dividing the fragments into four groups, depending on the first token:

1. Fragments starting with "— ", highest priority.
2. Fragments starting with "–".
3. Fragments starting with "-".
4. Other fragments, lowest priority.

The groups are ordered from the highest to the lowest priority. Within each group fragments are sorted according to increasing distance from the start of the sentence. For each fragment in this sorted list a search for the head noun is performed, until the noun is found or there are no fragments left.

[11] There are multiple rules used here and the main one detects full stops followed by capitalized common words.

2.3 Searching for the Head Noun

Searching for the head noun in the current sentence fragment is an algorithm that uses SFJP and traverses the tokens, comparing them to the *form specification* – a set of expected values of grammatical categories. The starting specification is *nominative noun*, but there are some exceptions, e.g. if the fragment prefix ends with a word from the second group, the head noun will be not a subject, but an object and the instrumental case is expected. There are some special token sequences that might change the process:

1. An adjective followed by a noun, both in the genitive case. Some Polish nouns in the singular genitive case look the same as in the plural nominative case, for example: “klasy” (*classes or class’*), “drzewa” (*trees or tree’s*). If they are preceded by an adjective, they can be disambiguated, so they will not be mistaken for plural. Example: “sredniej klasy” (*middle class’*).
2. A prepositional phrase. It never contains the head noun, so to prevent mistakes, for every noun the part of speech of the previous word is checked. If it is a preposition, the noun is ignored. Example: “z miasta” (*from a town*).
3. An adjective followed by a noun. Some Polish plural adjectives look the same as nouns. For example, the word “wloski” (*Italian*) in the expression “wloski malarz” (*Italian painter*) can be mistaken for a noun, meaning *trichomes*. It can be disambiguated by checking if it is an adjective matching to the form specification and if the following word is a noun that is in agreement with this adjective.

Another important issue is the frequent use of phrases like “jeden z ...” (*one of ...*) instead of the head noun. After such a phrase the plural genitive case should be expected. For example, instead of “najwiekszy zamek” (*the largest castle*) one can write “jeden z najwiekszych zamków” (*one of the largest castles*) or “najwiekszy z zamków” (*the largest of the castles*). To resolve this, we have to do an additional matching, checking if the current segment is an adjective matching the specification and followed by “z”. Not all adjectives can be used in that expression – allowed words include: “jeden”, ordinals (“pierwszy”, “drugi”, ..., “ostatni” - *first, second, ..., last*) and superlatives. If this matching succeeds, the form specification is changed to the plural genitive case and matching the gender of the adjective. The adjective is also saved in a list called *aux*, which contains auxiliary parts of the label.

2.4 Relations and Operators

The definition gives information about the category that described object belongs to. In other words, the object is often a kind of the category. Sometimes that relation is indicated explicitly:

Wellnhoferia

Wellnhoferia - **rodzaj** prehistorycznego ptaka blisko spokrewnionego z archeopteryksem.

The algorithm described above finds only the word “rodzaj” (*a kind*). The underlined part is the correct definition (*a prehistoric bird*) in the genitive case. To find the

Table 1. Relations and operators. The "number" column specifies expected grammatical number of the head noun after the operator. The expected case is always genitive.

Relation	Number	Operator examples
Rodzaj (*kind of*)	s. or pl.	gatunek, podgatunek, rodzaj, typ, forma, rasa, odmiana
Nazwa (*name of*)	s. or pl.	nazwa, okreslenie, tytul, oznaczenie
Czesc (*part of*)	s. or pl.	czesc, element, dzial, edycja
Zbiór (*set of*)	plural	grupa, rodzina, podrodzina, seria, lista, zbiór, gromada

correct head noun, we have to recognize the special word "rodzaj", which is an operator of the *kind of* relation. Then the form specification has to be switched to the genitive case and the search for the definition (the right side of the relation) should be continued. There are more operators of the *kind of* relation: "gatunek" (*species*), "typ" (*type*), etc. What is more, *kind of* is not the only relation type used in encyclopedic definition. Sometimes it is easier to define the object by describing it as a set of smaller objects: "Inuici – grupa ludów" (*Inuit – a group of tribes*) or a part of a bigger one. To find these relations and their operators, additional research was performed. It resulted in creating a list of relations and operators shown in Table 1. When an operator is found, it is added to the *aux* list and the form specification is updated to the appropriate number and the genitive case. Operators may also be chained together as well as with expressions of type "jeden z".

As an example, Figure 2 shows the steps of finding the head noun for an article about mayflies. The first sentence is: "Ecdyonurus – nazwa lac. jednego z rodzajów jetek." (*Ecdyonurus – the latin name of one of the genera of mayflies*).

Sometimes an operator is used as the head noun – this is a result of inherent natural language ambiguity. In that case the head noun will (hopefully) not be found. The algorithm takes the elements from the end of the *aux* list until it finds a noun. That noun becomes the head noun.

2.5 Additional Parts of Semantic Labels

There is a trade-off between the amount of additional information and conciseness of the semantic label. Because it is important to have a simple and well-defined label structure, after multiple experiments it was decided that there are only a few types of additional definition parts:

- a conjunction "i" (*and*) followed by a noun in the same form (number and case) as the head noun, e.g. "miasto i gmina" (*a town and a commune*),
- a noun in the genitive case, e.g. "ruda zelaza" (*iron ore*),
- a noun in the same form and gender as the head noun, e.g. "lekarz chirurg" (*a physician surgeon*),
- an adjective in the form matching the form and gender of the head noun, e.g. "chorwacki pilkarz reczny" (*a Croatian handball player*).

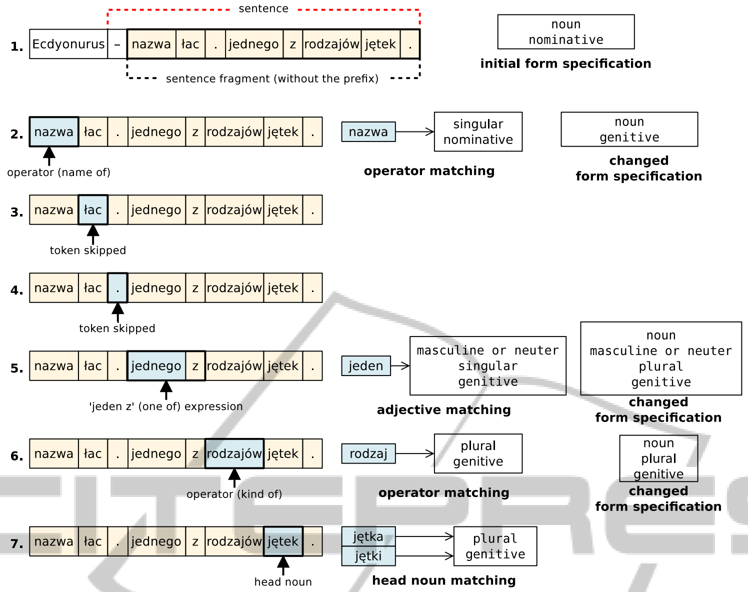

Fig. 2. Processing a complex definition with multiple operators. (1) Repeated headword is skipped, the first sentence is detected and a fragment is generated. The fragment prefix "-" is omitted. Initial form specification contains a noun in the nominative case. (2) The first token, "nazwa" (*name*) is an operator of the *name of* relation. The form specification is now changed to the plural genitive case. (3) The segment "lac" (abbreviation of *Latin*) does not match. (4) There is also no match for the dot. (5) Because of the expression "jednego z" (*one of*, masculine or neuter gender) the form specification is changed to the plural genitive case, masculine or neuter gender. (6) The word "rodzajów" (kinds, genitive case, plural) matches the specification. The form specification is changed again to the plural genitive case. (7) The word "jetek" (mayflies, genitive case) becomes the head noun (there are two words with this form: a feminie noun "jetka" and a plurale tantum word "jetki", this ambiguity cannot be resolved).

These additional parts may appear only directly after (any one of them) or before (only the last one) the head noun. The algorithm used for finding them uses form specifications to match the words. They are matched in the same order in which they were listed above[12]. It is important to note that adding these definition parts sometimes has a positive side effect of disambiguating the head noun. For example, the definition below contains an ambiguous head noun "polana", which can mean *a glade* or *logs* (plural).

```
Przyslopek
Przyslopek - nieduza polana i przelecz w Gorcach znajdu
jaca sie na grzbiecie laczacym Przyslopek (1123 m) z Ku
dloniem (1276 m). (...)
```

[12] This order performed best in tests, for example it allows for correct recognition of the expression "owoce morza" (*fruit of sea*), in which the token "morza" can be either a plural nominative or a singular genitive form of the word "morze" (*a sea*).

The definition means *Przyslopek – a small glade and a mountain pass in the Gorce mountains.* Matching the adjective “nieduza” (*small*) allows disambiguating the word “polana”. The full semantic label will be “nieduza polana i przelecz”.

2.6 Final Label Processing

The label requires some additional processing before it can be saved as a semantic label. The *aux* list contains adjectives that do not introduce much information. They are removed from the label. This generally does not result in loss of data except expressions like “jeden z najwiekszych zamków” (*one of the largest castles*), that will be changed to “najwiekszy zamek” (*the largest castle*), which has a slightly distorted meaning.

The form of the label has to be changed to the nominative case and the first noun after a removed adjective needs to be changed to singular if the adjective was singular. This may cause word ambiguity – for example the definition “jeden z rodów” (*one of the clans*) after processing becomes either “ród” (*a clan*) or “rod” (*rhodium*). If there is a need to have the definition in a short form without the operators, the *aux* list can be skipped. It results in a more concise albeit sometimes incomplete label.

After analyzing the output data statistics, it turned out that there are many headwords for which it is not needed to read the article to create a good semantic label. These include public institutions (schools, churches, airports, museums), administrative units (communes, nature reserves), valleys, vehicles, guns, flags and much more. These headwords are self-descriptive and the articles often contain no definition. To resolve this, a simple rule-based correction mechanism was developed. It contains a few hundred manually created rules that change the definition for the most common cases. It results in 1-2% accuracy gain.

3 Tests and Further Improvement

The tests were performed for Wikipedia data from the 20th of February 2010. There are 826117 pages. 636298 (77%) of them are unique articles and the rest are redirection pages. The definition extraction was performed only for the unique articles, because for each redirection the source semantic label is the same as the destination label. The number of headwords: 758423 is lower than the number of pages because of the existence of homonyms.

The goal of the first test was to find out how many of the articles contain correct definitions. It was difficult to perform this test automatically, so a sample of 500 random articles was created and manually checked. We can divide articles into three categories:

1. Correct definition. Articles that contain clear and correct definitions.
2. No definition. Articles without typical definitions – usually because the headword is self-descriptive.
3. No definition, no object. Tables, summaries, lists and other articles that do not describe a well-defined object. Example: “Formula 1 – Grand Prix Argentyny 1957”.

The results are shown in Table 2. Most of the articles contain correct definitions. The size of last two categories can be minimized by the rule-based correction algorithm. It helps to correct the definitions from the second category and delete the entries from the third one.

Table 2. Definition correctness for a random sample of 500 articles.

Category	Number	%
1. Correct definition	472	94.4
2. No definition	10	2.0
3. No definition, no object	18	3.6

The second test was a redirection check. It is good to know how accurate the redirections are and what is the probability that meanings of the source and the destination are identical or similar. A sample of 500 random redirections was created and manually divided into three categories:

1. Identical meaning. The perfect case.
2. Similar meaning. The meaning can be either slightly wider or narrower, or the number differs. Example: “Mewa” → “Mewy” (*Gull* → *Gulls*).
3. Different meaning. Example: “Perkusista” → “Perkusja” (*Drummer* → *Drums*).

The results are shown in Table 3. Most of the redirections have identical meaning. However, the usage of redirections might lower the definition quality by about one percent. If the quality is much more important than the amount of data and existence of redirections, they can be skipped.

Table 3. Differences in meaning for a random sample of 500 redirections.

Meaning	Number	%
1. Identical	478	95.6
2. Similar	10	2.0
3. Different	12	2.4

Table 4. Results of the semantic label accuracy test.

Category	Number	% of all (500)	Number of non-empty	% of non-empty
Correct	**452**	**90.4**	**444**	**92.5**
Incorrect (reasons below)	**48**	**9.6**	**36**	**7.5**
Error in redirection	2	0.4	2	0.4
No definition	5	1.0	3	0.6
No definition, no object (should be skipped)	5	1.0	5	1.0
Word not in SFJP	16	3.2	8	1.7
Other errors	20	4.0	18	3.8

The final and most important test is the semantic label accuracy test. It is often difficult to decide whether a semantic label is correct or not. However, for a given headword and first article paragraph it is quite easy to manually find the most suitable definition. This approach was utilized in the test. The first step was to select 500 random Wikipedia pages – redirections mixed with articles. For each of them a short and concise definition was manually selected. The main condition was that the definition should be based on the first paragraph of the article and on the title (both source and destination titles for redirections). Other words can be used only if there are no suitable ones in the article. If the article should be skipped, there should be an empty definition.

After preparing the test data the label extraction algorithm was run and the results were compared with manual definitions. The result for a single article was positive only if the automatically extracted label included the manually created one (only if all tokens were included in the same order). There were two test sets: one was used during the development and the other one for validation. The results for the validation set are shown in Table 4. We can view them from two different perspectives. The first one is the amount of correct definitions including empty definitions. It tells how many of the Wikipedia entries are processed correctly. The second one is the amount of correct definitions without empty definitions. It answers how many of the output dictionary entries are correct. Both values are over 90%, so the performance is good. There is no main reason of errors, but a frequent one is that the head noun is not in the SFJP dictionary. Other reasons include no definition in the article (sometimes also without a main object – so the definition should have been empty), errors in redirection, misspelling, too complex paragraph structure or other random coincidental mistakes.

The results shown above indicate that the algorithm can be still improved. It should be possible to skip articles without definitions and objects with better efficiency. SFJP could also be supplemented with additional frequent foreign words. There is also some room for improvement of the head noun detection algorithm, which fails in some complicated cases. Furthermore, it seems that the range of the search for additional label parts around the head noun could be extended – sometimes the head noun is correct, but other important definition parts are missing because they are separated from the head noun by other words.

Another issue is the further processing of headwords. The first token of a headword is always capitalized, what causes a need for a new algorithm that would decide about the case of this word. It could use the case of other words and the article contents to determine the right case.

4 Conclusions

Inflectional dictionaries are useful for natural language processing, but they lack semantic data. In this paper we investigated if such data can be easily obtained from an encyclopedia. We used the Wikipedia to create a dictionary containing semantic labels, describing the categories that headwords belong to. We described a method that uses the Polish Inflection Dictionary and several heuristics to extract a semantic label from the article: a short sequence of words containing the meaning of the headword. The labels are concise, have a formally defined structure and can be easily processed. Despite not using a predefined taxonomy or manual correction, the quality of output data is quite high. They may also be used to create a hierarchy of semantic categories, either manually or automatically.

The labels are going to be introduced into the Polish Inflection Dictionary. When this process is finished, it should be possible to assess the performance of the expanded dictionary. If we also connect it to SSJP, there should be a possibility to process Polish text using rich semantic information for the most common words and the labels for the less frequently used ones and proper names. For example, in the sentence “Blad pilota cessny byl glówna przyczyna katastrofy w Balicach” (Pilot error was the main cause of the disaster in Balice) the text processing algorithm would be able to know that “cessna” is a plane and “Balice” is an airport (using the semantic labels) and, after that, it could find the relations between *plane*, *airport* and *disaster* (using SSJP) and finally decide, that the sentence contains information about a plane crash. This kind of processing looks very promising and is a motivation for carrying on the research in this matter.

References

1. De Vel, O., Anderson, A., Corney, M., and Mohay, G. (2001). Mining e-mail content for author identification forensics. SIGMOD Rec., 30(4):55–64.
2. Edmonds, P. and Kilgarriff, A. (2002). Introduction to the special issue on evaluating word sense disambiguation systems. Nat. Lang. Eng., 8(4):279–291.
3. Fellbaum, C., editor (1998). WordNet: an electronic lexical database. MIT Press.
4. Gajecki, M. (2009). Słownik fleksyjny jako biblioteka jezyka c. In Słowniki komputerowe i automatyczna ekstrakcja informacji z tekstu. Wydawnictwa AGH, Krakow.
5. Kazama, J. and Torisawa, K. (2007). Exploiting wikipedia as external knowledge for named entity recognition. In EMNLP-CoNLL, pages 698–707. ACL.
6. Kuta, M., Chrzaszcz, P., and Kitowski, J. (2007). A case study of algorithms for morphosyntactic tagging of polish language. Computing and Informatics, 26(6):627–647.
7. Kuta, M., Kitowski, J., Wójcik, W., and Wrzeszcz, M. (2010). Application of weighted voting taggers to languages described with large tagsets. Computing and Informatics, 29(2):203–225.
8. Lubaszewski, W., Wróbel, H., Gajecki, M., Moskal, B., Orzechowska, A., Pietras, P., Pisarek, P., and Rokicka, T. (2001). Słownik Fleksyjny Jezyka Polskiego. Lexis Nexis, Kraków.
9. Medelyan, O., Milne, D., Legg, C., and Witten, I. H. (2009). Mining meaning from wikipedia. Int. J. Hum.-Comput. Stud., 67(9):716–754.
10. Milne, D. and Witten, I. H. (2008). Learning to link with wikipedia. In Proceedings of the 17th ACM conference on Information and knowledge management, CIKM ’08, pages 509–518, New York, NY, USA. ACM.
11. Pietras, P. (2009). Ekstrakcja leksykalna. In SÅ‚owniki komputerowe i automatyczna ekstrakcja informacji z tekstu. Wydawnictwa AGH, Kraków.
12. Pohl, A. (2009). SÅ‚ownik semantyczny jÄ™zyka polskiego. In SÅ‚owniki komputerowe i automatyczna ekstrakcja informacji z tekstu. Wydawnictwa AGH, Kraków.
13. Suchanek, F. M., Kasneci, G., and Weikum, G. (2008). Yago: A large ontology from wikipedia and wordnet. Web Semant., 6(3):203–217.
14. Toral, A. and Muñoz, R. (2006). A proposal to automatically build and maintain gazetteers for named entity recognition by using Wikipedia. In NEW TEXT - Wikis and blogs and other dynamic text sources, Trento.
15. Voorhees, E. M. (1999). Natural language processing and information retrieval. In Information Extraction: Towards Scalable, Adaptable Systems, pages 32–48. Springer, New York.
16. Wolinski, M. (2006). Morfeusz - a practical tool for the morphological analysis of polish. Advances in Soft Computing, 26(6):503-512.