The Semi-automatic Construction of the Polish Cyc Lexicon

Aleksander Pohl

Jagiellonian University, Cracow, Poland
aleksander.pohl@uj.edu.pl

Abstract. In this paper we discuss the problem of building the Polish lexicon for the Cyc ontology. As the ontology is very large and complex we describe semi-automatic translation of part of it, which might be useful for tasks lying on the border between the fields of Semantic Web and Natural Language Processing.
We concentrate on precise identification of lexemes, which is crucial for tasks such as natural language generation in massively inflected languages like Polish, and we also concentrate on multi-word entries, since in Cyc for every 10 concepts, 9 of them is mapped to expressions containing more than one word.

1 Introduction

The fact that linguistic resources play a key role in any Natural Language Processing undertaking is well established. Abstract theoretical problems such as word sense disambiguation and parsing as well as practical, such as machine translation, information extraction and question answering, are insolvable without large sets of fine grained rules, large semantic dictionaries or huge collections of hand-annotated texts.

When a researcher works on a language with only a few linguistic resources, she always has to decide, whether to create them from scratch employing the best available techniques or to adopt some of the already available lexicons, ontologies, etc. As the adoption of the WordNet lexical database [4] in the GlobalWordNet project shows, there is no obvious answer for this question.

Considering Polish, which is a language with a constantly growing set of linguistic resources (there are at least several complete or semi-complete Polish inflectional dictionaries, two growing WordNets and one large national corpus containing hand annotated samples of syntactic structures) one has to decide, whether it makes sense to wait for other researchers to complete their undertakings or to start the construction or adaptation of other resources.

Considering semantics, which is our primary field of interest, we have to agree, that the most advanced Polish resource is the Polish WordNet [8][1]. Since it is available for the Polish research community without restrictions and is created according to the state-of-the-art NLP techniques, it doesn't make sense to spend time and money, on the creation of another, similar resource.

[1] Available at http://plwordnet.pwr.wroc.pl/browser/?lang=en.

The Polish lexicon for Cyc [6], is a mapping between Cyc concepts and their Polish lexical representations. Since the mapping does not have to be isomorphic and each concept might have many mappings, the set of mappings for a give concept might be considered as a synset. What is more, the taxonomy of concepts in Cyc, in its structure, is quite similar to the taxonomy of WordNet synsets. At the first glance it seems, that the Polish lexicon is much similar to the Polish WordNet and, as a result, it seems to be a fruitless effort. Thus the question arises: what are the special properties of Cyc and what are the design goals of the Polish lexicon, which make the decision of creating it valuable?

2 Motivation

Our primary concern is to build algorithms and tools which bridge the gap between Polish language and the Semantic Web, thus bringing the benefits of the technology to the Polish speaking community.

Even though the fields of the Semantic Web and Natural Language Processing have much in common, there are certain problems, which have to be solved, before the data available in the Semantic Web and the data made available by NLP techniques is fully translatable. This stems from the fact, that the reference resources for the Semantic Web are ontologies, while the Princeton WordNet and its incarnations for languages other than English, serve as the *de facto* standard for NLP. Yet, there exist mappings between concepts of ontologies and WordNets (e.g. there is a mapping between Cyc and Princeton WordNet 2.0), but these mappings have certain limitation, stemming from the fact, that the logical structures of ontologies and WordNets is different.

The most problematic difference, in our opinion, is the huge discrepancy between the number and semantics of the types[2] of relations employed in both types of resources. In ontologies, the number of relations is not restricted *a priori* – it is only limited by the complexity of the domain of the ontology and by the desired level of detail. For instance, the old version of Dublin Core[3] defined 15 relations[4], while the latest defines approx. 50; the Music Ontology[5] defines approx. 120 relations, DBpedia[6] approx. 1200 and Cyc approx. 17000 relations[7].

On the other hand, most of the WordNets are created in accordance with the original Princeton WordNet idea refraining from using cross-part-of-speech relations. What is more, the set of relations was primarily limited to these, which were well accepted by the linguistic researchers community. Even though there are exceptions to these rules (e.g. there are cross-part-of-speech relations in the Polish WordNet), and there are plans

[2] From here, by relation we mean both type of a relation and instance of a relation. We hope this inadequacy will not introduce ambiguities, since in most cases the types of relations are discussed.

[3] http://dublincore.org/documents/dcmi-terms/

[4] In RDF/OWL oriented ontologies the relations are always binary and are called properties.

[5] http://musicontology.com/

[6] http://wiki.dbpedia.org/Ontology?v=zj4

[7] ResearchCyc, system: 10.126767, KB: 7141, http://research.cyc.com

and proposals to extend the set of relations (see [1]), it is unimaginable that the set of relations will grow to the size observed in moderately complicated ontologies.

To explain why we have to bother with that difference, let us consider a prototypical scenario, in which a music information extraction application utilizes data available both in the Semantic Web and made available by WordNet-based NLP algorithms. Let us assume, that the NLP module is able to fully disambiguate the common concepts (common nouns, verbs, adjectives, etc.) which appear in a certain text and the Semantic Web module is connected with a knowledge base containing massive amount of information about music[8]. The system should be able to answer questions such as „Have Tool already released the Ten thousand days album?", by parsing the question and consulting the database or recent press releases. However, it is unlikely, that the NLP module would recognize Tool as a name of a music group, and it is even less likely, that the phrase „Ten thousand days" would be recognized as a title of a music product, since the NLP dictionaries should capture general linguistic knowledge. But the biggest problem lies in the fact that the information is intransferable from the NLP module to the Semantic Web module – the former doesn't capture the relations between the release event (in which a music entity makes some music product available to the audience), the music entity and the music product. It might capture a notion of an event's actor and object, but such an information is too vague for the ontology.

We argue, that in such an application the NLP module should be designed in such a way, that the ontology contents is directly available in it. This is why we think, that building the Polish lexicon for Cyc, is worth its effort. The other advantages of using Cyc as the primary resource for NLP-enabled Semantic Web applications are as follows: there exists a Semantic Web endpoint which is linked to other Linked Open Data resources[9], it has probably the largest number of relations employed to describe the stored and processed knowledge, CycL – the language of Cyc is very expressive (e.g. allows for expressing relations between relations) and the ontology is shipped with an efficient inference engine, allowing not only for accessing, but also processing the knowledge in a consistent manner. And the last, but not the least, the relations in Cyc (and other ontologies) have formal definitions, which means, among others, that their arguments are restricted to concepts defined in the ontology (e.g. the first argument of the relation `#$weaponTypeCanDestroyTargetType` is restricted to `#$Weapon` and the second to `#$SolidTangibleThing`).

As it was stated, our primary concern is to bridge the gap between Polish language and the Semantic Web. Our final goal is to create a system, which is able to recognize ontological relations with their arguments in Polish texts, as well as, being able to produce well-formed Polish sentences, on the basis of the contents of the ontology. So, besides the adoption of a large number of relations provided by Cyc, we have to embrace the second important phenomenon – multi-word expressions. The reason why they are so common in ontologies stems from the fact, that the ontologies (and knowledge bases) contain two types of entities, which are mostly represented by multi-word expressions: proper names and „artificial" concepts.

[8] e.g. http://dbtune.org/musicbrainz/

[9] http://sw.opencyc.org

Proper names are the primary means for describing particular things, and as such they are quite valuable, since they might be used to automatically pick training examples for the relations. The „artificial" concepts are concepts which are used to properly structure the contents of the ontology – e.g. in Cyc there are concepts such as `#$Agent-Generic`, `#$Agent-PartiallyTangible` and `#$Agent-Underspecified`, which are used to capture certain properties of various types of agents. They shall not be mapped to the same word – *agens* – since that would introduce false ambiguity. It is better to provide descriptive, distinct mappings for these concepts (e.g. *agens*, *agens materialny*, *uogólniony agens*), but multi-word expressions are indispensable here. This is why we pay special attention to the multi-word expressions.

3 Related work

In our work we use both the transfer approach and the statistical approach to translate the contents of Cyc. The first method is used to translate the English names of the Cyc concepts, while the second method is used to find corresponding Cyc concepts, for semantic categories extracted from the Polish Wikipeida.

There has been much research in the field of statistical machine translation of the compound expressions (see [13]) and there are commercial machine translation systems available, like Google Translate[10]. On the other hand there is a lot of research concentrating on the extraction of the knowledge from Wikipeida (e.g. DBpedia [2], YAGO [14], WikiNet [7]).

Still there are two problems refraining us from directly using the tools and resources available so far. As for the statistics-base translation, we found out, that Google Translate is not well suited for the lexicon translation task. On the other hand, the lack of proper bilingual corpus did not allow us to utilise this method to the full extent. As for the resources derived from Wikipedia – although that most of the projects provides multi-lingual labels for the extracted concepts, there are two problems which have to be resolved. First of all – all the resources are based on the English version of the Wikipedia, which means that any Polish article not having its English counterpart, is not available there. Second – as Polish is an inflected language, the labels have to be accompanied with the precise inflectional information, which is not present in the above-mentioned resources.

4 Methodology

4.1 Goals

As it was stated in the Motivation section, our primary goal is to bridge the gap between Polish language, and the Semantic Web, using the Cyc ontology as the primary resource, by providing a relation extraction tool which is capable of recognizing Cyc relations in the Polish texts, and by generating Polish paraphrases for Cyc propositions. The first step to achieve this goal is to build the Polish lexicon for the ontology. At the first

[10] http://translate.google.com

glance, it seems that we should build a full lexicon, that is a lexicon covering all symbols available in Cyc. But we think, that to build and test the relation extraction system, this is not needed. We anticipate, that the precision of such a system will be correlated with the number of mappings, but it won't change dramatically, if only some of the concepts will be mapped.

This is due to the fact, that the proposed algorithm would utilize the definitions of relations, the argument constraints in particular. It appears, that only approx. 4 thousands of concepts are used as the argument constraints. We also observe, that there are certain meta-relations, such as `#$relationAllExists`, which could be quite useful, for the relation extraction task. The number of concepts appearing in these relations is approx. 8 thousands. That is why we propose to translate only *these* concepts and verify the feasibility of the information extraction algorithm construction.

Still, this assumption seems to be an oversimplification – even though we would be able to train the algorithm to recognize these relations with these concepts as their arguments on the basis of that mapping, we won't be able to recognize other concepts. E.g. we would be able to recognize the `#$releases-Underspecified` relation, in a sentence „Zespół wydał nową płytę" („A music group released a new album"), assuming that „music group" and „album" are the argument constrains of that relation, but we won't be able to recognize it in a sentence like „Tool wydał wczoraj CD Ten thousand days" („Tool released the Ten thousand days CD yesterday"), since „Tool", „Ten thousand days" and „CD" won't be recognized as the proper specializations of „music group", „album title", etc.

We agree, that this is a problem, but we won't resolve it by translating the full Cyc taxonomy. Instead, we are going to use the results of a project aiming at the extraction of the hyperonymy relation from the Polish Wikipedia, which was carried out in our research group [3]. In short, the results cover several hundreds of thousands of concepts, grouped within several thousands of semantic categories.

The particular goals we are going to achieve are as follows:

1. create translations for all the concepts which are used as the argument constraints in the Cyc relations (approx. 4 thousands)
2. create translations for part of the concepts which are appear in the meta-relations (approx. 2 thousands)
3. map these concepts to the semantic categories extracted from the Polish Wikipedia

Achieving these goals would allow us to:

1. automatically pick training examples for the Cyc relations
2. build linguistic models of these relations
3. build algorithms extracting these relations from Polish texts

The text generation feature of the designed system is not covered in this document, but the prototype applications utilizes it.

4.2 The algorithms

To build the most accurate mapping of the selected Cyc concepts, we decided to do it by hand. The tool presented allows for rapid construction of the lexicon, but does

not make the human operator unnecessary. As our previous research shown, the fully automatic translation is not feasible [10]. What is more, due to the ambiguity of the base forms of Polish words, for some of them it is not possible to automatically select their inflectional paradigms, which makes the human operator indispensable.

Thus the tool is primarily designed to facilitate the translation. This means, that it incorporates two translation algorithms (one transfer-based and one statistics-based) and other resources, such as the semantic categories extracted from Polish Wikipedia [3]. As an effect, if the proper translation and the proper mapping is suggested by the system, the human operator only verifies it. If not, all the data which is needed for the precise translation and mapping is presented to the operator, speeding-up the process.

4.3 The transfer based translation algorithm

The transfer based translation algorithm is used as a primary means for finding the proper translation and mapping for given Cyc concept. It works as follows – for each Cyc concept selected for the translation:

1. *translate* the English mapping of the concept into Polish (many results might be produced)
2. *map* the words of each translation to the entries of Polish inflectional dictionary
3. *transform* the translations to match syntax constraints
4. *rank* the translations
5. *present* the results to the human operator
6. *store* the selected result in the database
7. *search* for semantic categories extracted from the Polish Wikipedia, corresponding to the translation
8. *merge* or *link* the selected categories with the Cyc concept

Translation The first step in the creation of the Polish lexicon, is the translation or pairing of lexical units. This might be done by utilization of (in a transfer approach) a machine readable English-Polish dictionary or (in a statistical approach) a bilingual corpus. The latter approach is quite popular in the on-line translation systems, such as Google Translate and it has certain advantages – namely the translation algorithm is generic and the bilingual dictionary is not needed. Still, it seems that this approach is not well suited for the taxonomy translation task. It is due to the fact, that in most cases, the available training bilingual corpuses cover only texts containing regular sentences, having at least the SVO structure, while in most cases the Cyc concepts are described as a mere nominal expressions. What is more, the obtained translation should be canonical, that is, the head phrase should be in singular[11] and nominal case.

This prediction was verified on the Google Translate system. Out of 118 Cyc concepts, only 22 were translated exactly the same as by the human translator. 20 of them had certain syntax errors (e.g. „tangible agent" – „rzeczowe agent" where the adjective does not agree with the noun on case and gender), 40 of them had certain translation errors (e.g. „acquiring" – „przejmującej" where the concept denotes an event, while the

[11] or plural for *plurale tantum* nouns

translation is an adjective, thus a property), 64 of them were not in a canonical form (e.g. „animal" – „zwierząt" where the translation is in plural and in dative case, while it should be in singular and in nominal case), and 62 were translated differently, due to the general design principles of the lexicon (avoidance of ambiguities, among the others)[12].

This is why we choose the transfer-base approach as general means for providing the translations for the Cyc concepts. It is based on a large machine readable English-Polish dictionary „Wielki Słownik Multimedialny Polsko-Angielski/ Angielsko-Polski Oxford-PWN". Although the information which is available in such a dictionary is lexically rich – it signals the grammatical category of the entries, includes limited syntactical, semantical and pragmatical information – these features are not provided consequently and it is hard to obtain precise mapping between Cyc concepts and the dictionary entries on the one hand, and the translated entries and Polish inflectional dictionary entries on the other hand.

The translation strategy is as follows – when we translate some Cyc concept, which is represented by S_i^{en} character string, there might be the following general cases:

1. The character string is a single word, which *is not present* in the dictionary – we try to apply some transformation to it, such as stemming, but if the result is not present as well, we have to ignore it. If it is present, this situation is reduced to the next one.
2. The character string is a single word, which *is present* in the dictionary – we pass the list of translations $(S_{i,1}^{pl}, S_{i,2}^{pl}, S_{i,3}^{pl}, \ldots)$ to the next step of the algorithm.
3. The character string is a multi-word expression, which *has direct* representation in the dictionary – since it seems to be a compound expression, we process it if it was a single word – pass the whole list $(S_{i,1}^{pl}, S_{i,2}^{pl}, S_{i,3}^{pl}, \ldots)$ to the next step.
4. The character string is a multi-word expression, which *doesn't have direct* representation in the dictionary – we divide the S_i^{en} string into single words: $W_{i,1}^{en}$, $W_{i,2}^{en}, W_{i,3}^{en}, \ldots$, which might be represented by the following character strings $S_k^{en}, S_l^{en}, S_m^{en}, \ldots$ in the dictionary. Then we remove stop words (such as determiners or prepositions) form the list. For each element of the resulting list we take the corresponding Polish strings and create a vector of lists, where each position is occupied by the corresponding translations, and order of the list reflects the order of the source words: $\left[(S_{k,1}^{pl}, S_{k,2}^{pl}, \ldots)_1, (S_{l,1}^{pl}, \ldots)_2, \ (S_{m,1}^{pl}, \ldots)_3, \ldots\right]$. The lower index attached to parentheses indicates the position of the source word in the source expression. This vector is passed to the next step of the algorithm.

To sum-up – the translation of a Cyc concept produces a vector of Polish words or lists of Polish words, and since a single word might be considered as a single-entry list, we might simplify the description, by assuming, that always a vector of lists containing Polish words is produced, where each element of the vector corresponds to one word in the English mapping of the concept, and each element of the list corresponds to one possible translation of the word.

For instance: if we translate `#$AddictiveSubstance`, which is mapped to the `addictive substance` expression in English, we might receive the following result:

[12] The number of errors does not sum to 118, since one translation could be marked as invalid more than once.

$\big[$(uzależniający, wciągający)$_1$,(substancja, istota, ciężar, waga, podstawa, treść, realność, majątek)$_2\big]$

Mapping to inflectional dictionary Since the next step of the algorithm transforms the obtained translations to match the syntax constrains of the Polish language, the result of the previous step has to be mapped to Polish inflectional dictionary, such as one described in [16] or [9]. This dictionary should have at least two features:

1. lemmatization – recognition of the lemma based on any of the inflected forms
2. inflection – production of an inflected form based on a provided set of tags

Since the first feature might introduce ambiguity (e.g. the character string *goli* is an inflected form of lexemes having the following lemmas: *gol* (goal), *golić* (to shave), *golić się* (to shave oneself), *goły* (naked), *Gola* (a Polish surname) and *Goły* (a Polish surname)), for each character string $S^{pl}_{i,j}$ we might receive many lemmas:

$$S^{pl}_{i,j} \rightarrow \left[L^{pl}_a, L^{pl}_b, \ldots\right]_{i,j} \tag{1}$$

where L^{pl}_a stands for the lexeme with an index a. The i, j indices indicate, that given vector corresponds to the $S^{pl}_{i,j}$ Polish character string.

The indexing of the lexemes in the dictionary needs some special attention – in general we would like to avoid the situation in which human interpretation of each lexeme requires looking it up in the index, so the lexeme should be at least represented by its lemma.

As it is discussed in detail in [15], there are no better means of differentiating the lexemes with the same lemma and different inflection, than by introducing some arbitrary marking of the homonymuous lemmas. [15] proposes numbering of the lemmas, while we think that the approach proposed in [9], namely attachment of inflectional label to each lemma, is better, since, provided that the person who looks at the mapping, knows the labeling system, she does not have to check the ordering of these lemmas to determine, what is the inflectional paradigm of the lexeme, represented by the $< lemma, inflectional\ label >$ pair[13].

In the system described in [9] the inflectional label consist of capital letters and is constructed in such a way, that the most significant morphological distinctions are placed at the beginning of the label, thus the first letter determines the grammatical category of the lexeme (A – noun, B – verb, etc.) the second letter (in the case of nouns) determines their gender and so on. This idea has another advantage, that only the lemma, the inflectional label and the inflectional scheme is necessary to produce all the forms of a given lexeme, without the direct intervention of the dictionary, so it's easier to port across operating systems and versions of the dictionary.

In fact, the label could be replaced by a number, but the most important difference between theses systems is that, in the first case, the index indicates the position of the lexeme among other lexemes with the same lemma, while in the second case, it

[13] We have to mention, that in our formalization of lexemes $L^t_a, L^t_b, \ldots$ we keep abstract indices a, b, which should be interpreted as distinct $< lemma, inflectional\ label >$ pairs.

indicates position of its inflectional paradigm among other inflectional paradigms. This means that in the second case, the index is less likely to change.

Since we assumed that the output produced by the previous step is always a vector, the result of this step is a vector of mappings obtained by merging the vectors produced for a given position in a given single or multi-word entry:

$$\left[(S^{pl}_{k,1}, S^{pl}_{k,2}, \ldots)_1, (S^{pl}_{l,1}, \ldots)_2, (S^{pl}_{m,1}, \ldots)_3, \ldots\right] \rightarrow \tag{2}$$

$$\left[\left([L^{pl}_a, L^{pl}_b, \ldots]_{k,1}, [L^{pl}_c, \ldots]_{k,2}, \ldots\right)_1,\right.$$

$$\left.\left([L^{pl}_d, \ldots]_{l,1}, \ldots\right)_2, \left([L^{pl}_e, \ldots]_{m,1}, \ldots\right)_3, \ldots\right] \rightarrow \tag{3}$$

$$\left[(L^{pl}_a, L^{pl}_b, L^{pl}_c, \ldots)_1, (L^{pl}_d, \ldots)_2, (L^{pl}_e, \ldots)_3, \ldots\right] \tag{4}$$

where the vector $[L^{pl}_a, L^{pl}_b, \ldots]_{k,1}$ is merged with the vector $[L^{pl}_c, \ldots]_{k,2}$ producing the list present at the first position of the equation 4.

We have to mention that, some of the elements of the character string lists might be removed completely, when they are not recognized by the inflectional dictionary. Due to the productive character of languages, there are always some words, which are missing in such dictionaries. For example, the latest version of the dictionary described in [9] doesn't recognize forms such as *konfigurować* (configure) or *opcjonalny* (optional), which are recognized by modern general purpose Polish dictionaries such as the online version of the most popular Polish dictionary accessible on the web-site `http://sjp.pwn.pl`[14].

We also have to say that there are character strings, which are not recognized by the dictionary due to the fact, that they are multi-word expressions. It's because some English words might be translated as Polish multi-word expressions. In such a case we have two options: to ignore them or to split them into single words. In our case we took the first approach in cases, when given Polish translation corresponded to one word in an English multi-word expression and the second approach in the opposite cases.

Considering the example from the previous step, the vector would be translated into the following result: $\Big[$(<uzależniać,BDA>, <uzależniać się,BDA>, <uzależniający,CAA>, <uzależniający się,CAA>, <wciągać,BDA>, <wciągać się,BDA> <wciągający,CAA>, <wciągający się,CAA>)$_1$, (<substancja,ADACBAA>, <istota,ADAAA>, <ciężar,ACAAAAA>, <Ciężar,AAAAD>, <waga,ADAB>, <Waga,AABACC>, <podstawa,ADAAA>, <treść,ADCCA>, <realność,ADCCA>, <majątek,ACABA>)$_2\Big]$

Transformation The mapping step might produce tens or even hundreds of interpretations for a single Cyc concept, thus some transformations have to be applied, to reduce these numbers. There is also another problem, which stems from the fact, that for the

[14] Checked on the 7^{th} of March 2010.

translation to be consistent, certain features of the lexemes (such as gender of a noun and an adjective) have to be accommodated.

The general idea of the transformation step, is to look at the grammatical categories of the lexemes corresponding to the Polish character strings. It is observed, that for resources such as Cyc, many of the source entities were two-word expressions or three-word expressions containing a preposition or a determiner. Because determiners are not present in Polish, and preposition often disappear in the translation (e.g. „of" is replaced by the genitive case of the dependent nominal phrase), we had to deal with restricted number of grammatical category combinations and it was quite easy to order them in an effective, yet not much restricting manner:

1. noun + adjective
2. noun + noun
3. noun + verb
4. noun + other
5. other

Having such an ordering, the lexeme tuples taken from the Cartesian product of the vector from the equation 4, were partitioned into five sets and only the non-empty partition with the highest rank was selected. All the other lexeme pairs were dismissed. We haven't defined rules for triples of lexemes, since it turned out, that such complex expressions were rarely translated correctly, and they were processed rather slowly.

After this reduction, the inflectional forms of the lexemes were adjusted to fulfil Polish syntactic rules. For the first 3 cases the schema was as follows:

1. noun + adjective: the base form tagging for the noun was determined, which could be a nominal case of a singular or a plural form (the latter for *plurale tantum* nouns), then the form of the adjective was selected accordingly (its number, case and gender being taken directly from the noun form and its gender).
2. noun + noun: for each of the nouns the number was determined as in the previous rule. Then two pairs of forms where added: in the first, a nominal case for the first and genitive case for the second noun was selected, in the second – a genitive case for the first and nominal for the second lexeme[15].
3. noun + verb: the infinitive form of the verb and the accusative case[16] for the noun were selected.

In the other cases only the base forms were selected.

When these transformations had been completed the number of lexemes' pairs were significantly reduced. But what is even more important, for most of the cases, the obtained expressions were grammatically correct.

This step would transform the example from the previous step as follows:

```
[uzależniająca substancja, uzależniająca się substancja,
```

[15] This idea was supported by the fact, that in nominal phrases consisting of two nouns, the subordinate noun, always has the genitive case, e.g. *wąsy kota* (whisker of a cat) – `plural:nominal singular:genitive`. In other words: noun governs genitive case.

[16] Since the syntactic features of Polish verbs are not yet fully described in a form of an electronic dictionary, we've taken this assumption, although it could produce many incorrect translations.

```
wciągająca substancja, wciągająca się substancja, uzależniająca
istota, uzależniająca się istota, wciągająca istota, wciągająca
się istota, uzależniający ciężar, uzależniający się ciężar,
wciągający ciężar, wciągający się ciężar, ...]
```

Ranking When the transformation step is finished, the translations are ranked according to the number of their occurrences in a large corpus. In general, three different cases might appear as the result of the previous step:

1. there are only single words in the vector
2. there are pairs of lexemes produced by the transformation step
3. there is the original vector from the equation 4, if it contained more than two positions

In the first case, the results are ranked simply according to the number of their occurrences in the corpus. It is quite important, that there is one big lemmatized corpus of Polish available for free – the IPI PAN corpus described in [11].

In the second case, the results are ranked according to the number of occurrences of bigrams in the corpus. Even though the IPI PAN corpus is not well balanced, the mere fact that given pair of words occurs in it, is sufficient to properly order them and reject uncommon multi-word translations.

In the last case, when the complex transformation was not applied to the original vector, the translations should be ranked as in the first case, but separately for each position. It is not practical to check the number of occurrences of each combination of the character strings, as it might be really huge, and would significantly slow down the translation process (while the whole methodology is devised for its speed-up). Nevertheless, the ordering of translations for each position is still important, since it provides the human operator with translation hints and also signifies, which translation might be the most natural.

It might seem that the IPI PAN corpus is too small for such a task, and tools such as search engine should be consulted. In practice, this is not the best idea, since the number of queries which would have to be send to the server is quite large (at least tens for single concept), and as the tool is designed for interactive usage, this would slow down the process – simple looking through the list of results would be more efficient[17].

For the example provided above, only the „substancja uzależniająca" is recorded in the corpus and only this proper translation is presented to the user.

Selection When the translations are ranked, they might be presented to the human operator. If the number of translations is too large, they might be cut at some level (e.g. only 15 top ranked translations appear), since it doesn't make sense to go through all of theme (this might be more time consuming than figuring out the translation from scratch).

[17] On the other hand, if the results were cached, the search-engine approach would be much better and it is considered for the further development of the system.

The selection should be as easy as clicking a button next to the correct translation. We think that the user should not be disrupted by the precise lexical information at the moment, so simple character strings should be displayed. This might introduce some ambiguity, but since the user always should have the option to enter the translation manually[18], the interpretation step is necessary anyway.

Interpretation The last necessary step in creating the accurate translation is the correct interpretation of the character string selected by the user. Although if he selects one of the translations that was suggested by the system, the necessary information is available at hand. But when he enters the translation manually, the correct lexemes and taggings should be selected.

This is done with a support of a simple parsing algorithm. There is not enough space to discuss it in detail, that's why we give just its simple characteristic. In general the algorithm is based on the concept of unification, with features attached to the grammatical categories [5, pages 489–528]. Each grammatical category defines what values of features are required from the other grammatical categories if their instances are subordinate elements of the instances of the former category in the abstract syntax tree (e.g. genitive case for the noun which is a subordinate of some other noun). Some of the categories are supplemented with the information, that their instances require obligatory subordinate elements (like reflexive pronoun *się* for reflexive verbs). For given interpretation of the expression – if there exist tree of nodes constructed according to the optional requirements, and for each node all of its obligatory requirements are met, the interpretation is marked as valid.

Still, although for simple expressions, this algorithm produces many unambiguous results, there are cases[19], for which even the most clever algorithm would produce more than one interpretation. So in such cases, the human operator should be able to select the correct lexemes and taggings by hand.

As a final result, the Cyc concept is mapped to the list (in the simplest case – single entry list) of Polish lexemes with taggings[20] attached:

$$L_i^s \rightarrow \left[(L_a^t, T_{a,k})_1, (L_b^t, T_{b,l}), \ldots\right] \tag{5}$$

Searching for semantic categories Since we have the proper mapping of the Cyc concept, we might search for the semantic categories extracted from the Polish Wikipedia, which are most similar to the translation. So far this algorithm is not much complicated – all the entires which contain the lexemes which appear in the translation are selected, and then they are ranked according to the following equation:

$$R_i = \frac{cm_{i,j}}{cl_i} * \frac{cm_{i,j}}{cl_j} * children_i \tag{6}$$

[18] The rationale is that there are many cases, like compound metaphoric expressions not recorded in the bilingual dictionary, or entries containing many words (in our case, more than two) that will never appear as the result of the complex processing.

[19] E.g. the lexemes with lemma *zamek* differentiated by the `singular:genitive` *zamka*:*zamku* or the expression *akt własności*, where the second lexeme might be in singular or in plural.

[20] Indicating the selected forms.

where

1. R_i – is the rank of semantic category i
2. $cm_{i,j}$ – is the number of common lexemes in the semantic category i and the translation of the Cyc concept j
3. cl_i – is the number of lexemes in the name of the semantic category i
4. cl_j – is the number of lexemes in the translation of Cyc concept j
5. $children_i$ – is the number of instances in the semantic category i

This equation favors categories with many instances on the one hand, and also the categories, whose names are similar to the translation of the concept on the other.

Merging and linking the categories As the last step of the algorithm the user might make the following decisions:

1. She might merge zero or more semantic categories with the Cyc concept. The result of the operation will be a direct attachment of all the instances of the category to the Cyc concept – the instances of the category will become the instances of the concept. E.g. the „miasto" (city) category might be merged with the `#$City` concept.
2. She might link zero or more semantic categories with the Cyc concept. In this case the category will become a specialization of the concepts, and its instances will be treated as indirect instances of the concept. E.g. the „miasto wojewódzkie" (provincial city) might be linked with the `#$City` concept.

The user is not restricted to merging one category, due to the fact, that the semantic category extraction algorithm produces many over-specified categories (e.g. „miasto położone" (city situated), „miasto znajdujące" (city situated), which should be merged with the `#$City` concept).

Even though the previous step might produce many results, it is not needed to merge or link all of them – the less instances given semantic category contains, the less time should be spend for its analysis and some of them might be simply skipped.

4.4 The statistics based translation algorithm

Due to the fact, that the automatic construction of the Polish lexicon is merely feasible and that there is no free English-Polish bilingual corpus, we decided to employ a statistics based translation algorithm, not in the full extent, but only for the data, which is well suited for that. Instead of using it to translate the Cyc concepts, due to the way the Polish semantic categories are extracted from the Polish Wikipedia, it turned out that such an algorithm might be used to easily find the proper mappings for these categories, especially those, which have the highest number of instances.

The semantic categories extraction algorithm described in [3], is designed to extract the hyperonymy relation from the definitions of the articles. It does not take into account the Wikipedia categories assigned by users (like the WikiNet project [7]), nor does it take into account the infoboxes (like the DBpedia project [14]). Instead it performs shallow parsing of the definitions, trying to figure out the most probable location of

the *genus proximum* of the concept. This feature of the algorithm is used, first – to find the most probable English translation of given semantic category, second – to find the most probable Cyc concept, which corresponds to it. The English translation is determined, by exploiting the interlingual links present on the Wikipedia sites. Although most of projects aimed at extracting the knowledge from this encyclopedia, takes them as 100% accurate (e.g. DBpedia, YAGO, WikiNet), we observed that for the articles about abstract concepts, such as species, this is not true (consider the link between the Polish word „osa" and English „wasp" – in English it stands for a biological order, but in Polish it stands for a particular species).

On the other hand, if there are many Polish articles with given semantic category assigned connected with their English counterparts, the noise might be weed out by the statistics. What is more important – the more the category has instances, the better the results should be, and as an effect it is quite easy to cover the largest categories bringing the formal semantics of Cyc to many Polish expressions.

The statistics based algorithm works as follows:

1. For each Polish article find the corresponding *English article* via the interlingual link. If the link is not present, skip this article in further processing.
2. For each English article corresponding to the Polish article, extract the *hypernym* of the concept being described, by simple pattern matching algorithm.
3. When the Polish semantic category is selected for the *translation*, determine it by analyzing the English semantic categories (hypernyms) of English articles corresponding to the Polish articles covered by the category.
4. For the most probable English translations of the category, determine the *Cyc concepts* corresponding to them.

Finding the English article corresponding to the Polish article The first step of the algorithm is straight-forward. It might be further simplified, if instead of using the raw Wikipedia data, the pre-processed data is downloaded directly from the DBpedia download page[21] (it contains the contents of the knowledge-base split into several files covering different informations, such as the interlingual links and the definitions extracted from the articles).

Extracting the English semantic categories Following the methodology described in [3] and adapting the hypernym extraction patterns described in [12], we used the patterns in table 1 to determine the candidate semantic categories for given English article.

The extracted expressions are further processed to remove additional information given in parentheses and the group-marking expressions (*group*, *series*, *species*, *type*, etc.). The end of the category is detected as an occurrence of a preposition, a form of the verb *to be* or a dot. Since there might be more than one semantic category in the matched expression, it is split in places where conjunctions and determiners appear. As a result, for each article a list of semantic categories might be extracted.

[21] `http://dbpedia.org/download351`

Table 1. Semantic category extraction patterns

Regexp format	Example
X are (an?|the) Y	Bloc Party are a British...
X (is|was) one of the Y	Dubai is one of the seven...
Xs are Y	Bees are flying insects...
X is an?(\S* of)? Y	Hornbills are a family of bird...
X is the Y	Anthropology is the study of humanity...
X (was|were) (an?|the) Y	Kipchaks were an ancient Turkic...

Translation of the Polish semantic category The translation of a given Polish semantic category works as follows: for those instances of the category (i.e. Polish articles from the Wikipedia, this category was extracted for) having their English counterparts, build a histogram of the English semantic categories extracted from these English articles with the number of their instances as the value. Then for each semantic category whose name is one word longer than the original category, produce all the combinations of the length same as the name of the original category. The original categories are replaced with these combinations, and the number of instances of these categories is divided by their length in words (since for each category exactly the same number of combinations is produced). Then the remaining categories with names longer than the name of the original category plus one, are removed from the histogram, but each category with name of the same size is treated as a pattern, and the instances of the removed categories matching this pattern are added to that category. Finally five English categories with the largest number of instances are selected as the candidate translations of the Polish category.

Selection of the Cyc concepts corresponding to the Polish category As a last step of the algorithm, the Cyc concepts corresponding to the Polish category are selected. These concepts are obtained via the `denotation_mapper` Cyc call. Still, the results of the call might be ambiguous, so another histogram for the Cyc concepts is build. It is used to determine the final ranking of correspondence between the Polish categories, and the Cyc concepts. The value assigned to a given concepts is the sum of values assigned to its English translations divided by the number of all concepts returned by the `denotation_mapper` call. As a result, a list of concepts sorted by their relevance is presented to the user.

5 Application

As a part of the research we constructed an application[22], along the lines of the described methodology. In our earlier research we found out, that although rough auto-

[22] The demo of the application is available under the URL `http://klon.wzks.uj.edu.pl/cycdemo`.

Fig. 1. Main window of the application.

matic translation of single-word entries produces promising results, the syntactic information, which is indispensable for the natural language production capability, has to be entered by the human operator. What is even more important, most of the entries which represent concepts of the ontology are multi-word expressions, which are mistakenly translated by the leading statistics based machine translation systems, like the above mentioned Google Translate.

The default mode of operation of the application is as follows. First, the list of concepts selected for the translation is loaded to the application. Then the human operator logs into the application, and starts the interactive session. The list of concepts is paginated for easier operation and by default is ordered by the their names. When the user selects the concept for a translation, the system provides him with suggestions derived by the transfer based translation algorithm. If one of the suggestions is valid, the user selects it. If there is no valid translation, the user might provide the proper translation by hand. Then the translation might be further validated, by consulting the Polish paraphrases based upon Cyc knowledge and the translation itself. If there is an error in the translation, the user might delete it, and provide new, valid translation. If it is correct, the user might search for the Polish semantic categories corresponding to the translated concept. If there are any, the user might merge the or link them with the concepts. If not, the user goes to the next concept.

The main window of the system is presented on figure 1 and it shows part of the list of concepts which are mapped to Polish expressions (the concept is on the left, while the translation is on the right). The user might get familiar with the meaning of the concept,

Fig. 2. Translations suggested by the application.

Fig. 3. Morphosyntactic ambiguity.

by studying its generalizations (up arrow on the left), specializations (down arrow on the left) and directed instances (dot icon on the left) and by reading its comment (the left icon in the middle between the concept and its translation). The number next to the arrows and the dot indicates the size of the corresponding set. The letters above the list of the concepts are used to filter them by their first letter, while the numbers are used to move from one page to another. The concepts might be also sorted by their name, number of subtypes, and number of instances (links just above the concepts list).

When the user clicks the right icon which is between the concept and the translation, the user sees the translations suggested by the system (Fig. 2). The English mappings of the concept are present at the top of the translation box. The manual entry box and the suggested translations are below. The user might accept given suggestion by clicking the plus icon next to it or enter some other translation in the manual entry box. The user might also consult statistics of given expression by clicking the `bigramy` link next to it. The caption left to the translation indicates, if is was produced by the compound translation algorithm (*pwn full*) or was found directly in the dictionary (*pwn*).

In the rare case, that given expressions is morphosyntacticly ambiguous, the user is consulted once again (Fig. 3). The full tagging of each lexeme is presented, as well as the morphological information in a form of an inflectional label. If the user is not familiar with the inflectional scheme, he might click the lemma of the lexeme, and its full inflectional paradigm will be presented. The user accepts given interpretation by clicking the tick icon, which is below.

When the translation is provided, the user might validate his selection by clicking the icon which is to the left of the translation (Fig. 1). It will show Polish paraphrasing

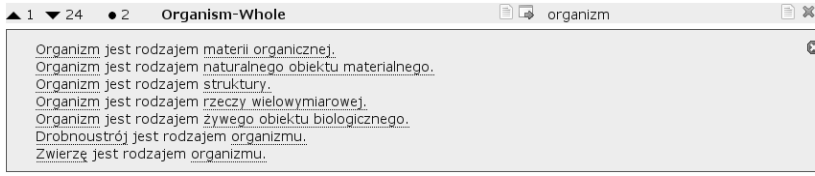

Fig. 4. Polish paraphrases of sample of the Cyc knowledge.

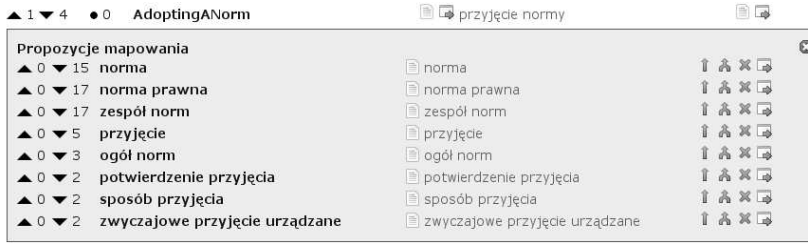

Fig. 5. The semantic categories which are proposed for merging and linking.

of sample of the taxonomical knowledge taken from Cyc (Fig. 4), which utilizes the morphosyntactic information attached to the translation. If the user hovers over the underlined expression, he will see the corresponding Cyc concept or the Polish semantic category.

When the user is sure, that the translation is valid, he might search among the semantic categories extracted from the Polish Wikipedia, to find these which could be merged or linked with the concept (Fig. 5). This is done by clicking on the icon rightmost to the translation.

The user might merge given category with the concept, by dragging and dropping the split arrow on the name of the concept. As a result, all the instances of the category will be linked with the concept, and the category will be removed. The user might also link the category with the concept, by dragging and dropping the straight arrow on the name of the concept. As a result, the category will be linked with the concept by the generalization relation.

The alternative mode of operation of the application allows for finding Cyc concepts corresponding to the Polish semantic categories with many instances. It is available when the user clicks the `pojęcia` section in the main menu (Fig. 6). The mode of operation is similar to the standard mode, and the main difference is that the list contains Polish semantic categories, which are ordered by number of instances (Fig. 7). The user goes through the list and tries to find the Cyc concept, which is most similar to the category. The statistics based translation algorithm produces the suggestions, and if there is a similar Cyc concept, the category might be merged or linked with that concept. If the algorithm did not provide any meaningful suggestions, the user might further

Fig. 6. The main menu of the application.

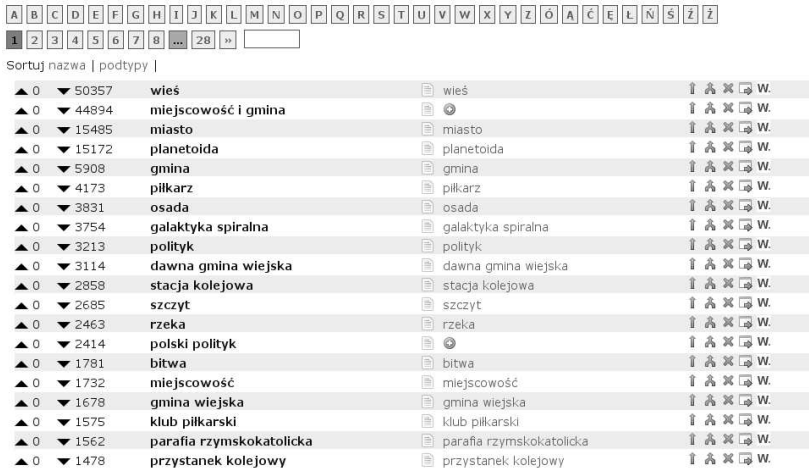

Fig. 7. The list of Polish semantic categories ordered by number of instances.

search for the corresponding Cyc concept, by utilising the generic search functionality (available under the name `szukaj` in the main menu – Fig. 6).

The main view of the alternative mode of operation is presented on the Figure 7 and it shows the list of Polish semantic categories, that the Cyc concepts are sought for. If the meaning of the name of the category seems to be ambiguous, the user might check the extenstion of the category (i.e. all the Wikipedia articles, which the category was extracted for), by clicking the down arrow left to it. The number next to the arrow indicates the size of the category. The categories might be navigated the same way as the Cyc concepts.

As with the Cyc concepts, the lexicalization of the semantic category is separated from the category itself, to allow multiple lexicalizations. The name of the category is on the left, while the top-most lexicalization is on the right. The user might check the lexicalizations and provide new, if he clicks the default lexicalization.

The Cyc concepts found by the statistics based translation algorithm corresponding to the category are presented, when the user clicks the right-most icon, next to the default lexicalization of the category („W” icon on the Figure 7). A list of Cyc concepts sorted by their relevance is then presented to the user. The mechanism for estabilishing

Table 2. Performance of the transfer-based translation (estimated over 600 concepts).

	Precision	Recall	F-measure
Google Translate	18.6%	100%	31.4
Transfer-based translation	37%	88%	52

the correspondence between the categories and the concepts is the same as in the default mode of operation. The only difference is that the position of the categories and the concepts is swapped.

6 Results

So far 2479 of Cyc concepts out of approx. 6 thousands selected for the translation were mapped[23] to the Polish expressions. The translations were carried out by two independent translators, reaching the inter-translator agreement of 56%[24], which means that the translation task is not easy. This is due to the fact, that the concepts selected for translations are mostly general, and have to be translated carefully.

The precision[25] of the transfer-based translation was 37% and recall[26] was 88%. The precision for the two-word compounds was 27%. A comparison with the Google Translate is given in the table 2.

The results for the statistics-based translation, in terms of finding the Cyc concepts corresponding to the Polish semantic categories, was substantially better. The raw precision[27] of the method (i.e. the Polish to English expression correspondence) was 69%[28] and the recall[29] was 89%. The final precision[30] of the method (i.e. the Polish category to Cyc concept correspondence) was 92%[31] and the recall[32] was 95%.

The performance of the method is in terms of semantically-related translations/concepts is given in table 3 (for Polish-English expression correspondence) and 4 (for category-concept correspondence).

[23] Within approx. one month period.
[24] Estimated over 600 concepts.
[25] Measured as the number of concepts for which the system suggested the translation, which was then selected by the translator.
[26] Measured as the number of concepts for which translations were suggested.
[27] Measured as the number of correct translations divided by the number of categories for which the translations were provided by the algorithm.
[28] Estimated over 98 categories.
[29] Measured as the number of categories for which the translations were provided by the algorithm, divided by the total number of categories evaluated.
[30] Measured as the number of Cyc concepts selected as corresponding to the categories, divided by the total number of categories for which the Cyc concepts were provided by the algorithm.
[31] Estimated over 166 categories.
[32] Measured as the number of categories for which Cyc concepts were suggested, divided by the total number of categories evaluated.

Table 3. Raw performance of the statistics-based translation (estimated over 98 categories).

	Precision
Exact translation	69%
Including super-concepts	86%
Including sub-concepts	73%
Including overlapping-concepts	80%

Table 4. Final performance of the statistics-based translation (estimated over 166 categories).

	Precision
Exact mapping	92%
Including super-concepts	95%
Including sub-concepts	93%
Including overlapping-concepts	95%

As a final result 534 of Cyc concepts were mapped to Polish semantic categories, extracted from the Wikipedia. These categories cover approx. 220 thousands of Polish articles, which seems to be a good result for the short amount of time spent on the translation and mapping.

7 Conclusions

Although the precision of the transfer-based translation algorithm is quite low, the application speeds-up the creation of the Polish lexicon for Cyc. This is due to the fact, that it integrates several resources, namely the Cyc ontology, Polish inflectional dictionary, English-Polish dictionary as well as semantic categories and concepts extracted from Wikipedia, while presenting to the user only these pieces of information, which are relevant for the task. As it was expected, the statistics-based translation algorithm performed substantially better, but its scope was limited due to the absence of a proper bilingual corpus.

It is estimated, that after few months (approx. 3) of work the created resource will cover thousands of Polish linguistic units incorporated into the formal framework of the Cyc ontology.

The usefulness of this resource is verified in experiments covering the extraction of semantic relations from Polish texts as well in demo application allowing for Polish paraphrasing of the knowledge available as Open Data. The preliminary results are of these experiments are promising.

References

1. Amaro, R., Chaves, R.P., Marrafa, P., Mendes, S.: Enriching Wordnets with new Relations and with Event and Argument Structures. In: Seventh International Conference on Intelligent Text Processing and Computational Linguistics. pp. 28 – 40 (2006)
2. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Ives, Z.: DBpedia: A nucleus for a web of open data. Machine Translation 14(2), 113–157 (2005)
3. Chrząszcz, P.: Automatyczne rozpoznawanie i klasyfikacja nazw wielosegmentowych na podstawie analizy haseł encyklopedycznych. Master's thesis, UST, Cracow (2009)
4. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press (1998)
5. Jurafsky, D., Martin, J.H.: Speech and language processing (second edition). Prentice Hall (2009)
6. Lenat, D.B.: CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM 38(11), 33–38 (1995)
7. Nastase, V., Strube, M., Börschinger, B., Zirn, C., Elghafari, A.: WikiNet: A Very Large Scale Multi-Lingual Concept Network. In: Proceedings of the Seventh conference on International Language Resources and Evaluation, (LREC'10) (2010)
8. Piasecki, M., Szpakowicz, S., Broda, B.: A Wordnet from the Ground Up. Oficyna Wydawnicza Politechniki Wrocławskiej (2009)
9. Pisarek, P.: Słowniki komputerowe i automatyczna ekstrakcja informacji z tekstu, chap. Słownik fleksyjny, pp. 37–68. Uczelniane Wydawnictwo Naukowo-Dydaktyczne AGH (2009)
10. Pohl, A.: Automatic Construction of the Polish Nominal Lexicon for the OpenCyc Ontology, pp. 51–64. EXIT (2009)
11. Przepiórkowski, A.: The potential of the IPI PAN corpus. Poznań Studies in Contemporary Linguistics 41, 31–48 (2006)
12. Sarjant, S., Legg, C., Robinson, M., Medelyan, O.: "All You Can Eat" Ontology-Building: Feeding Wikipedia to Cyc. In: Web Intelligence'09. pp. 341–348 (2009)
13. Somers, H.: Review Article: Example-based Machine Translation. Machine Translation 14(2), 113–157 (2005)
14. Suchanek, F., Kasneci, G., Weikum, G.: YAGO: A Large Ontology from Wikipedia and WordNet. Web Semantics: Science, Services and Agents on the World Wide Web 6, 203–217 (2008)
15. Woliński, M.: System znaczników morfosyntaktycznych w korpusie IPI PAN. Polonica XII, 39–54 (2004)
16. Woliński, M.: Morfeusz – a Practical Tool for the Morphological Analysis of Polish. In: Intelligent Information Processing and Web Mining, IIS:IIPWM'06 Proceedings. pp. 503–512,. Springer (2006)