

The Polish Cyc lexicon as a bridge between Polish language and the Semantic Web

Aleksander Pohl

Computational Linguistics Department,
Jagiellonian University, Cracow, Poland
Email: 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.

I. 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 set of fine grained rules, large semantic dictionaries or huge collections of hand-annotated texts.

When a researcher works on a language having much less available resources than English, 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 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 [1]¹. Since it is available for the Polish research community without restrictions and is created according to the state-of-the-art techniques of building WordNets, it doesn't make sense to spend time and money, on the creation of another, similar resource.

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

The Polish lexicon for Cyc [2], 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?

II. 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 types² 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 Core³ defined 15 relations⁴, while the latest defines

²From 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/>

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

approx. 50; the Music Ontology⁵ defines approx. 120 relations, DBpedia⁶ approx. 1200 and Cyc approx. 17000 relations⁷.

On the other hand, most of the WordNets is 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 and proposals to extend the set of relations (see [3]), 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 an ontology containing massive amount of information about music⁸. 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 resources⁹, 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 for not only 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 contain two types of entities, which are mostly represented by multi-word expressions: proper names and „artificial“ concepts.

Proper names are the primary means for describing particular things, and we think that they are quite valuable, since they might be used to automatically pick training examples for the relations, we have to embrace them. 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 materialny*, *uogólniony agens*), but multi-word expressions are indispensable here. This is why we pay special attention to the multi-word expressions.

III. RELATED WORK

In our work we use the transfer approach to translate the compound expressions. On the other hand there has been much research in the field of statistical machine translation of these expressions (see [4]) and there are commercial machine translation systems available, like Google Translate¹⁰. The problem we encountered, refraining us from using this approach, was the lack of resources – we didn't find any publicly available aligned English-Polish corpus, while using resources such as Wikipedia seemed to be too demanding for this task. What is more – direct application of Google Translate (which is discussed later) was too erroneous.

IV. METHODOLOGY

A. 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 tool which is capable of recognizing its relations in Polish texts, and by generating Polish paraphrases for them. The first step to achieve this goal is to build the Polish lexicon for the ontology. At the first glance, it seems that we should

⁵<http://musicontology.com/>

⁶<http://wiki.dbpedia.org/Ontology?v=zj4>

⁷ResearchCyc, system: 10.126767, KB: 7141, <http://research.cyc.com>

⁸e.g. <http://dbtune.org/musicbrainz/>

⁹<http://sw.opencyc.org>

¹⁰<http://translate.google.com>

build a full lexicon, that is a lexicon covering all symbols available in Cyc, but we think that this is not needed. 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 and we propose to translate only *these* symbols (it might appear in the future, that translating more symbols, would bring better extraction results).

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 constraints 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 [5]. 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 translation for all the concepts which are used as the argument constraints in the Cyc relations (approx. 4 thousands)
- 2) map these concepts to the semantic categories extracted from the Polish Wikipedia

Achieving these two 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 utilize it.

B. The algorithm

The general algorithm for building the Polish lexicon is as follows – for each Cyc concept which is used as an argument constraint:

- 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

C. Translation

The first necessary 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 singular¹¹ 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 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 approach, 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 really hard to obtain precise mapping between Cyc concepts and the dictionary entries on the one hand, and 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 trans-

¹¹or plural for *plurale tantum* nouns

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

formation 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 ($S_{i,1}^{pl}, S_{i,2}^{pl}, S_{i,3}^{pl}, \dots$) 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 ($S_{i,1}^{pl}, S_{i,2}^{pl}, S_{i,3}^{pl}, \dots$) 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: $W_{i,1}^{en}, W_{i,2}^{en}, W_{i,3}^{en}, \dots$, which might be represented by the following character strings $S_k^{en}, S_l^{en}, S_m^{en}, \dots$ in the dictionary. Then we remove stop words (such as determiners or prepositions) from 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: $\left[(S_{k,1}^{pl}, S_{k,2}^{pl}, \dots)_1, (S_{l,1}^{pl}, \dots)_2, (S_{m,1}^{pl}, \dots)_3, \dots \right]$. 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: $[(uzależniający, wciągający)_1, (substancja, istota, ciężar, waga, podstawa, treść, realność, majątek)_2]$

D. Mapping to inflectional dictionary

Since the next step of the algorithm will transform 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 [6] or [7]. This dictionary should have at least two functionalities:

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

Since the first functionality 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

Goly (a Polish surname)), for each character string $S_{i,j}^{pl}$ we might receive many lemmas:

$$S_{i,j}^{pl} \rightarrow [L_a^{pl}, L_b^{pl}, \dots]_{i,j} \quad (1)$$

where L_a^{pl} stands for the lexeme with an index a . The i, j indices indicate, that given vector corresponds to the $S_{i,j}^{pl}$ 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 [8], there are no better means of differentiating the lexemes with the same lemma and different inflection, than by introducing some arbitrary marking of the homonymous lemmas. [8] proposes numbering of the lemmas, while we think that the approach proposed in [7], namely to attach an inflectional label to each lemma, is better, since, provided that the person who looks at the mapping, knows the labeling system, doesn't have to check the ordering of these lemmas to determine, what is the inflectional paradigm of the lexeme, represented by the $\langle lemma, inflectional label \rangle$ pair¹³.

In the system described in [7] 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 indicates position of its inflectional paradigm among other inflectional paradigms. This means that in the second case, the index is much 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 given single or multi-word entry:

$$\left[(S_{k,1}^{pl}, S_{k,2}^{pl}, \dots)_1, (S_{l,1}^{pl}, \dots)_2, (S_{m,1}^{pl}, \dots)_3, \dots \right] \rightarrow \quad (2)$$

$$\left[\left([L_a^{pl}, L_b^{pl}, \dots]_{k,1}, [L_c^{pl}, \dots]_{k,2}, \dots \right)_1, \left([L_d^{pl}, \dots]_{l,1}, \dots \right)_2, \left([L_e^{pl}, \dots]_{m,1}, \dots \right)_3, \dots \right] \rightarrow \quad (3)$$

¹³We have to mention, that in our formalization of lexemes L_a^t, L_b^t, \dots we keep abstract indices a, b , which should be interpreted as distinct $\langle lemma, inflectional label \rangle$ pairs.

$$\left[(L_a^{pl}, L_b^{pl}, L_c^{pl}, \dots)_1, (L_d^{pl}, \dots)_2, (L_e^{pl}, \dots)_3, \dots \right] \quad (4)$$

where the vector $[L_a^{pl}, L_b^{pl}, \dots]_{k,1}$ is merged with the vector $[L_c^{pl}, \dots]_{k,2}$ 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 [7] 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: [*<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,ACAAAA>*, *<Ciężar,AAAAD>*, *<waga,ADAB>*, *<Waga,AABACC>*, *<podstawa,ADAAA>*, *<treść,ADCCA>*, *<realność,ADCCA>*, *<majątek,ACABA>*]₂

E. 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 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 were 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 lexeme¹⁵.
- 3) noun + verb: the infinitive form of the verb and the accusative case¹⁶ 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 example from the previous step as follows: [*uzależniająca substancja*, *uzależniająca się substancja*, *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*, ...]

F. Ranking

When the transformation step is finished, the translations are ranked according to the number of their occurrences in a large corpus.

¹⁵This 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.

¹⁶Since 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.

¹⁴Checked on the 7th of March 2010.

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 [9].

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 general case, if ngrams were employed in the transformation step, any kind of smoothing or backoff algorithm could be employed (see [10, pages 83–122] for details).

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 sent 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 efficient¹⁷.

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

G. 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 them (this might be more time consuming than figuring out the translation from scratch).

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

¹⁷On 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.

might introduce some ambiguity, but since the user always should have the option to enter the translation manually¹⁸ the interpretation step is necessary anyway.

H. 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 [10, 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 cases¹⁹, 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 taggings²⁰ attached:

$$L_i^s \rightarrow [(L_a^t, T_{a,k})_1, (L_b^t, T_{b,l}), \dots] \quad (5)$$

I. 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 entries which contain the lexemes which appear in the translation are selected, and then they are ranked according to the following equation:

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

where

¹⁸The 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.

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

²⁰Indicating the selected forms.

- 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.

J. 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.

V. APPLICATION

As a part of the research we constructed an application²¹, along the lines of the described methodology. In our earlier research we found out, that although rough automatic 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.

²¹The demo of the application is available under the URL <http://klon.wzks.uj.edu.pl/cycdemo>.



Fig. 1. Main window of the application.



Fig. 2. Translations suggested by the application.

The main window of the system is presented in Fig. 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, by studying its hypernyms, hyponyms and directed instances (icons on the left) and by reading its comment (the left icon in the middle between the concept and its translation).

When he clicks the right icon in the middle of the table, he 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. He might also consult statistics of given expression by clicking the *bigramy* link next to it.

In the rare case, that given expressions is morphosyntactically 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.

The user might validate his selection by clicking the icon which is on the left of the translation (Fig. 1). It will show Polish paraphrasing of sample of the taxonomical knowledge taken from Cyc (Fig. 4), which utilizes the morphosyntactic information attached to the translation.

Then the user might search among the semantic categories extracted from the Polish Wikipedia, to find these which could



Fig. 3. Morphosyntactic ambiguity.



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

be merged or linked with the concept (Fig. 5). This is done by clicking the icon on the right of the translation.

The user might merge the category with the concept, by dragging the splitted arrow and dropping it on the name of the concept. The user might also link the category with the concept, by dragging the straight arrow and dropping it on the name of the concept.

VI. RESULTS

So far 560²² of Cyc concepts out of 3600 selected for translations were mapped to the Polish expressions. The translations were carried out by two independent translators, reaching the inter-translator agreement of 56%. This means that the translation task is not easy. This is due to the fact, that the concepts selected for translations are quite general, and have to be translated carefully. The general precision²³ of the translations suggested by the system was 37% and recall²⁴ was 88%, while precision for two-word compounds was 27%.

On the other hand 193 of the translated concepts were linked and merged with Wikipedia semantic categories, covering more than 20000 Wikipedia instances (articles), which seems

²²It took two weeks to translate these concepts. The translation of the rest of them will be started in September 2010, since the funds for the task are paid in tranches.

²³Measured as the number of concepts for which the system suggested the translation selected by the translator.

²⁴Measured as the number of concepts for which translations were suggested.



Fig. 5. The semantic categories which are proposed for merging and linking, to be a good result for the very limited resources available for the project.

VII. CONCLUSIONS

Although the precision of the 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 a result, 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 and the preliminary results are promising.

REFERENCES

- [1] M. Piasecki, S. Szpakowicz, and B. Broda, *A Wordnet from the Ground Up*. Oficyna Wydawnicza Politechniki Wrocławskiej, 2009.
- [2] D. B. Lenat, "CYC: A large-scale investment in knowledge infrastructure," *Communications of the ACM*, vol. 38, no. 11, pp. 33–38, 1995.
- [3] R. Amaro, R. P. Chaves, P. Marrafa, and S. Mendes, "Enriching Wordnets with new Relations and with Event and Argument Structures," in *Seventh International Conference on Intelligent Text Processing and Computational Linguistics*, 2006, pp. 28 – 40.
- [4] H. Somers, "Review Article: Example-based Machine Translation," *Machine Translation*, vol. 14, no. 2, pp. 113–157, 2005.
- [5] P. Chrzęszcz, "Automatyczne rozpoznawanie i klasyfikacja nazw wielosegmentowych na podstawie analizy haseł encyklopedycznych," Master's thesis, UST, Cracow, Poland, 2009.
- [6] M. Woliński, "Morfeusz – a Practical Tool for the Morphological Analysis of Polish," in *Intelligent Information Processing and Web Mining, IIS:IIPWM'06 Proceedings*. Springer, 2006, pp. 503–512..
- [7] P. Pisarek, *Słowniki komputerowe i automatyczna ekstrakcja informacji z tekstu*. Uczelniane Wydawnictwo Naukowo-Dydaktyczne AGH, 2009, ch. Słownik fleksyjny, pp. 37–68.
- [8] M. Woliński, "System znaczników morfosyntaktycznych w korpusie IPI PAN," *Polonica*, vol. XII, pp. 39–54, 2004.
- [9] A. Przepiórkowski, "The potential of the IPI PAN corpus," *Poznań Studies in Contemporary Linguistics*, vol. 41, pp. 31–48, 2006.
- [10] D. Jurafsky and J. H. Martin, *Speech and language processing (second edition)*. Prentice Hall, 2009.