Natural Language Understanding as a classification process: report of initial experiments and results

Pedro Mota, Luísa Coheur, Ana Mendes

L^{2}F / INESC-ID Lisboa
Lisboa Rua Alves Redol, 9, 1000-029 Lisboa, Portugal

Abstract. In this paper, we show how Natural Language Understanding can be modeled as a classification task, regardless of the desired output format (an utterance, a logical formula or a frame). Named entities (and synonyms) can be easily incorporated in the whole process. Several scenarios are used in our experiments, corresponding to the different typical outputs of Natural Language Understanding systems: in the first, given an utterance, the system is supposed to identify the most similar utterance from a pre-defined database of utterances; in the second, the system aims at mapping a given utterance into a logical form; in the third, the system creates a frame from a given utterance.

Keywords: Natural Language Understanding, classification, semantic representation

1 Introduction

The task of Natural Language Understanding (NLU) aims at mapping utterances into some sort of representation that models their meaning and that the computer is able to digest. From a panoply of formalisms, these representations can be sentences [14], (quasi-) logical forms [4] or frames [17]. Different systems, depending on their goals and domains, use different semantic representations; also, the mapping between an utterance and the corresponding semantic representation (that is, the process of understanding) can be done in many different manners [3, 4, 7, 15, 20].

One way of approaching NLU is to model it as a classification task. However, this process needs to be adapted to the representation chosen by the user. In this paper, we show that, regardless of the output format (a sentence, a formula or a frame), understanding can be implemented as a classification task. The underlying module only needs to be fed with a training file and it learns how to interpret unseen utterances. Training files are (small) sets of interaction pairs of the form `(utterance, semantic representation)`, and, contrary to other approaches [18], there is no need to relate the semantic representation with parts of the utterances. Some basic features are already implemented and dictionaries containing named entities or synonyms can also be easily added to the process. As a consequence, the resulting module allows the rapid development of baselines by non experts, since the syntax of the knowledge sources does not force any kind of expertise. In fact, if one wants to rapidly test an understanding module – in a limited domain and ignoring a possible dialogue context – this can be a solution.

In this paper, we focus on three case studies that represent different output formats. In the first, we use a corpus obtained from an agent that operates as a museum guide; given an utterance, the system is supposed to identify the most similar sentence from a pre-defined database of utterances. In the second, the system targets to map a given utterance into a logical form; the used data is taken from a natural language interface to a cinema database. In the third scenario, the system should create a frame from a given utterance; the corpora used in this experiment is from an agent that operates in a domotic environment. It should be noticed that, in all the scenarios, a small amount of data is used for training, although accuracy results are generally above 80%. In addition, we also study the impact of incorporating a named entity recognizer in the understanding process and the impact, in frames classification, of previously determining the frame type itself. We intent to make our system and the used corpora available, allowing it to work as a baseline/benchmark.

The paper is organized as follows: in Section 2 we present related work, in Section 3 we formalize understanding as a classification process, and, in Section 4, we detail our experiments. Finally, in Section 5 we draw some conclusions and point to future work.

2 Related Work

ELIZA [20] is one of the first references in NLU. ELIZA used regular expressions to match the user input against pre-defined patterns that were responsible to trigger an answer. In this way, there was no complex representation involved and the target was an utterance that could be sent back to the user as a response. In fact, logical forms have been the representation target of most language understanding modules – such as natural language interfaces to databases [19] – since the early days. In many of those systems, logical forms – usually subsets of first order logic – were used as intermediate representation languages, facilitating the further mapping into the final representation (for instance, SQL). One of the systems that better details the target representation and the mapping process – based on syntax-semantic correspondences – is the Core Language Engine, fully described in [4].

Much progress was done in the understanding process and a panoply of different approaches exists. Starting with the agents' community that target communication through Natural Language, many agents use in-house NLU modules, some based on keyword spotting [9], some more complex, based on natural language processing modules [5, 10]. The process of understanding is also the key point of other applications, such as chatbots and dialogue systems. Considering the former, these continue with ELIZA tradition and usually follow a simple pattern-based approach, sometimes allowing the use of other knowledge sources, such as synonyms. However, they are usually limited to small domains; alternatively, instead of feeding in particular information sources, they target to look "human". Thus, they are not worried in providing effective information to an user request, but in providing a plausible response (*Why do you want to talk about this?*). With respect to dialogue systems, there are many available, most of them providing rule-based NLU modules. State of the art references are Olympus [7], Trips [3] and Plow [2] as they provide impressive possibilities for the development of dialogue systems.

Contrary to the strictly rule-based approaches, at the present time many systems use machine learning approaches to NLU [17]. A possible way to model NLU can be made through the statistical processing of data. Here, although being given a sequence of words *W*, the target is still to determine the semantic representation of its meaning *M*. This is done through the maximization of $P(W|M)$, as Equation 1 shows:

$$\hat{M} = argmax_M P(M|W) = argmax_M P(W|M)P(W) \tag{1}$$

From Equation 1 two distinct models emerge: the *semantic prior model* and the *lexicalization model*. The former assigns a probability to a semantic structure; the later assigns $P(W|M)$ to a sentence *W* given the semantic structure. This paradigm can be applied to systems that require the use of frames, as described in [18]. To accomplish this, the definition of the following concepts is required:

- A set of segments $S = (s_1, ..., s_n)$. These segments correspond to the semantic structure $M = q_1, ..., q_n$ of the frame.
- An alignment $A = (a1, ..., a_n)$, that assigns a certain segment to a particular frame slot.
- A length $L = (l1, ..., l_n)$ for the segments.

The inclusion of these new variables results into a reformulation of the *lexicalization model* probability assignment: $\sum_{S,A,L} P(W, S, A, L|M)$. An Hidden Markov Model (HMM) can be used in this approach, in which variables *S*, *A* and *L* are hidden variables. The observable events in the Markov chain correspond to the words that compose the sentence *W*.

The best advantage of the statistical approach is the fact that it does not require building a grammar, making the adaptation to any set of data very easy. The major weakness of this method is the data-sparseness problem, which can be solved with a large number of example sentences.

Another approach within the machine learning paradigm treats understanding as a classification process. [6] tests several classification methods in order to obtain frames from a given utterance, and focus on the advantage of attaining interesting results with a small training dataset, although not much details are given about this dataset. This idea is further explored in a statistical approach called Cross Language Model (CLM) [15]. The goal of this approach is to give a set of question answer pairs in order to train the system. After the training process the system is able to receive an utterance, classify it and return the associated answer. This is done by comparing the probability distributions of words in the question and answer sets. The problem with this technique is that there might be correct question answer pairs that do not have any words in common. In order to overcome the vocabulary mismatch problem, it is assumed that there are two distinct languages, one for the questions and another for the answers. Therefore, answers need to be translated into the question language, or the other way around, before comparing the word distributions. The CLM approach has been used with success in a virtual agent called Sargeant Blackwell, which plays the role of a military instructor that is prepared to talk about the US Army and answers the questions of potential recruits [11]. An implementation of the CLM is available in the NPCEditor[1] toolkit [13].

[1]NPC stands for Non Player Character.

In [16] there is another example of a NLU system based on classification, returning utterances and logical forms as the representation target. However, to our knowledge, there is no available system that combines different outputs in a classification-based approach. This paper is our response to this gap.

3 Modeling understanding as a classification process

In this section we explain how, given different representation targets, the process of mapping utterances into these representations is modeled as a classification process.

3.1 Targeting similar utterances

In some particular domains it can make sense to encode the agent's knowledge as a set of utterances associated with the possible answers they will trigger. This is what is done, for instance in [12] and can be a very practical approach if the goal is to quickly develop and deploy a system in order to collect real utterances (that is, utterances from real users). This has also the advantage of not requiring experts to build grammars, since the input is simply a way of relating utterances with possible answers (as in the following example).

```
<interaction>
  <utterances>
     <u>Hello</u>
     <u>Good morning</u>
  </utterances>
  <answers>
      <answer>Hello, my name is Edgar.
      </answer>
  </answers>
</interaction>
<interaction>
  <utterances>
      <u>What do you do in the palace?</u>
      <u>What is your job at the palace?</u>
  </utterances>
  <answers>
      <a>I am the butler of the palace.</a>
  </answers>
</interaction>
```

With this approach, understanding implies choosing which of the utterances is the most similar with the one posed by the user. In order to model understanding as a classification task, we implement the strategy presented in [16], where authors automatically associate a virtual category to each interaction. For instance, considering the previous example, `CAT_1` can be associated with the first interaction and `CAT_2` with the second one, resulting in the following training corpus, as well as in a corpus that relates categories with answers:

Training corpus:
`CAT_1` *Hello.*
`CAT_1` *Good morning.*
`CAT_2` *What do you do in the palace?*
`CAT_2` *What is your job at the palace?*

Corpus relating categories with answers:
`CAT_1` *Hello, my name is Edgar.*
`CAT_2` *I am the butler of the palace.*

The classification process is then straightforward and, at the end, an answer with the category that was given to an utterance has to be selected.

It should also be noticed that, from such corpus, the set of most discriminative words can also be extracted, for instance, by using a decision tree, where each word present in the corpus is seen as an attribute. The resulting list of words can also be used to simulate a keyword spotting approach, since it is probable that real users will not interact with the system with paraphrases of the known utterances.

3.2 Targeting Frames

A frame is a set of slots, and a slot is an attribute/value pair. In order to model the mapping of a given utterance into a frame as a classification process, we should have a training set built from a set of utterance/frame pairs. For example, consider the utterances *Switch off the light of the living room, please* and *Switch on the television*, and the attributes {`where`, `dispositive` and `action`}. Here, the attribute `where` can take the values `living room` and `room`, the attribute `dispositive` can take the values `light` and `tv` and, finally, the attribute `action` can take the values `switch on` and `switch off`. Consider also, that all the attributes can take `NIL` as a value. A possible training set is:

```
<Frame>
  <utterances>
    <u>Switch off the light of the living room</u>
  </utterances>
  <attribute name="where">
    <value>living room</value>
  </attribute>
  <attribute name="dispositive">
    <value>light</value>
  </attribute>
  <attribute name="action">
    <value>switch off</value>
```

```
  </attribute>
</Frame>
<Frame>
  <utterances>
    <u>Switch on the television</u>
  </utterances>
  <attribute name="where">
    <value>NIL</value>
  </attribute>
  <attribute name="dispositive">
    <value>TV</value>
  </attribute>
  <attribute name="action">
    <value>switch on</value>
  </attribute>
</Frame>
```

Modeling the mapping of utterances into frames as a classification process involves training several classifiers (as many as the involved attributes) and, then, combining the obtained values. That is, each attribute will result in a classification model, that targets to classify each utterance with the possible values of that attribute. Thus, for each attribute, the system is trained in order to learn how to associate the correct value to that attribute, according with the associated utterance. For instance, considering the attribute `where`, the system will learn that the utterance *Switch off the light of the living room, please* will be classified as `living room` and that the utterance *Switch on the television* will be classified as `NIL`; by the same token, for the attribute `dispositive`, the first utterance will be classified as `light` and the second utterance as `TV`. The major problem with this approach is to combine the attributes, as the classifier can return high values for attributes that are not related with the given utterance. [6] presents several heuristics to combine the attributes, observing that these are dependent on the application domain in hands.

Another hypothesis is to previously give a category to each frame type and then to train a model to map the utterances into the frame type. For instance, consider that both previous frames had type DISPOSITIVE and that there was another frame of type FOOD for interactions with requests related with food. Then, according to the type of the frame, several models are trained in order to associate values to each attribute of the frame, as previously. Notice that the problem of having attributes not related with the frame no longer applies. We test both approaches.

3.3 Targeting Logical Formulas

When systems target logical forms, these are usually much simpler than, for instance, first order logical formulas. The formulas are often primitive formulas, that is, N-ary predicates with N terms as arguments. For instance, a possible representation to the question *Who directed Casablanca?* could be `QT_WHO_DIRECTED(casablanca)`. In this scenario, the developer of the system has to decide which are the predicates to be considered and their number of arguments, and which terms can appear in which argument.

In fact, these formulas can be seen as frames. Considering the previous example, the frame would be identified as `QT_WHO_DIRECTED` and the attribute/value pair would be `arg1: casablanca` (its only slot).

It should also be noticed that if we are able to include a dictionary with named entities (with entries like, for instance, `MOVIE` *Casablanca*), the problem could be reduced to the classification of the involved predicate plus a dictionary-based named entity recognition process. That is, the system needs to be able to associate the predicate `QT_WHO_DIRECTED` to the utterance *Who directed Casablanca?*, and afterwards, to detect that *Casablanca* is a `MOVIE`. The named entity should then be added to the predicate as its argument.

4 Experiments

In this section we test the classification process in three different scenarios, each one representing a different understanding situation. We also show results regarding different features and, in particular, the improvements we gain by detaching in-domain words in the important word/expressions list.

4.1 Experimental setup

The involved scenarios are the following:

- in the first scenario, we use the corpus of an agent that operates in a museum, answering questions about art, the ART corpus. Here, pairs relating utterances with possible answers are given to train the system;
- in the second scenario, we use a corpus collected during the development of a natural language interface from a cinema database, with real users, from now on the CINEMA corpus. Concerning this corpus, a set of questions and the respective semantic representation are given. Moreover, we use a named entities dictionary, which is mandatory, due to their variability;
- in the last scenario, we use the DOMOTIC corpus – that is a corpus in the domotic domain, which intends to support an agent operating in a house of the future. In this corpus a set of utterances and associated frames are used for training the classifiers.

Some details about these corpora are given in Table 1[2]. It should be noticed that the number of the target categories represents: a) for the ART corpus, the number of different interactions; b) for the CINEMA corpus, the number of involved predicates; c) for the DOMOTIC corpus, the different types of frames involved. In addition, within this late corpus there is a total of 18 distinct attributes (4 appear in multiple frames) and 93 possible values. Also, the lowest number of attributes in a frame is 2 and the highest is 4.

[2] In what concerns the CINEMA corpus we are excluding movies and actors' names from the counting.

	ART	CINEMA	DOMOTIC
Utterances	943	232	470
Total Words	5521	1645	3386
Unique Words	546	175	441
Categories	130	24	16

Table 1. Characteristics of three different corpora used in the experiments

Regarding the classification algorithm, we use Support Vector Machine (SVM) – the `LIBSVM` [8] implementation, with the most appropriate kernels and optimized parameters, using the one-versus-all multi-class strategy – although other classifiers can be used. As features we used a unigram version of the corpus and also a list of important words/expressions was manually built for each domain. If an utterance contains one of such words/expressions, then we append it to the utterances of the same category that do not have such word/expression. The sizes of these lists used in the ART, CINEMA and DOMOTIC corpus are 326, 28 and 44, respectively.

In all the experiments presented in the next section, we perform a 5-fold cross validation, where each corpus is split in 5 and then, the classification process is ran 5 times: in each time 80% of the corpus is used for training and the remaining 20% for test. We opted for this number of partitions because the used corpora has a relatively low number of training utterances, thus, if a higher number of partitions is used the test partition would end up with few test utterances in each category.

4.2 Results

ART corpus In what concerns the ART corpus, the first experiments were conducted in order to determine the best features. Best results were attained with a combination of unigrams and the (manually built) important words/expressions list. Then, since creating a manual list of important words is time consuming, we wanted to see if it would be possible to use an automatically created list in the classification process. A decision tree was used to find the most discriminative words in the corpus and the 60 most discriminative words were extracted. Then, the same best features obtained with the manual crafted list were used. Results can be seen in Table 2.

Features	Accuracy
Unigrams+Manual List	0,86 ±0,05
Unigrams+Automatic List	0.85 ±0,05

Table 2. Comparing manual vs. automatically built lists of important words/expressions

The results obtained in this experiment can be due to the fact that the training and test corpus share many paraphrases. That is, in the ART corpus there are many ways of saying the same thing, some resulting from simple reordering of words. Thus, if one sentence is in the training file and its syntactical paraphrase in the test file, it is almost certain that the classifier will be able to correctly tag it.

It is curious to see that using an automatically created list of 60 words as a feature results in almost the same accuracy as by using a manual built list with 326 entrances.

CINEMA corpus Here, as said before, our goal was to study the impact of having a pre-processing step with a named entity recognizer, that labeled as `ACTOR` and `MOVIE` the names of the actors and movies presented in the corpus (respectively), thus reducing the data sparsity. It should be noticed that all actors and movies in the training corpus had a corresponding entry in the named entities dictionary. Tests were made with unigrams and the manually built list of important words/expressions. Results are shown in Table 3 and, as expected, the introduction of the named entity recognizer had an important impact on the results.

NE recognition	Accuracy
Yes	0,82 ±0,07
No	0,71 ±0,07

Table 3. Results for the CINEMA corpus

DOMOTIC corpus In the first experiments with the DOMOTIC corpus, we have previously classified each frame. This can guide the next step, because we can identify to which frame an utterance belongs to and, thus, we are able to understand which slots need to be filled. In the training of classifiers that are used to fill the slots, utterances that do not contain any value for the respective attribute are discarded. We call to this experiment Frame First (FF). The second experiment follows the work from [6]: we train the system in order to teach it to give values to each attribute; then, we combine the different attributes in order to create a frame. It should be noticed that in this second experiment an utterance that does not have a value for an attribute is not discarded, and the category `CAT_UNKNOWN` is assigned to that attribute instead. In what concerns the combination of attributes, we choose the ones that attain the best scores. These scores are calculated based on the average of the scores returned by each individual attribute classifier for that frame. If an attribute returns the `CAT_UNKNOWN` category, then the individual score will be 0. After having all scores, the best one is chosen and the corresponding frame and attributes are returned. It should be noticed that when doing this calculation, the difference between the scores of two candidates may be very low. In these cases the tie breaker is the number of recognized categories (the ones that are different from `CAT_UNKNOWN`). We call Attributes First (AF) to this second experiment. Results of FF and AF are presented in Table 4 and were obtained with the same features used before (unigrams plus important words/expressions list). The same features as before have been used in both experiments.

	FF Algorithm	AF Algorithm
Frame Accuracy	0,96 ±0,04	0,78 ±0,03
Attribute Accuracy	0,92 ±0,03	0,81 ±0,05

Table 4. Results for the DOMOTIC corpus.

The overall results of all experiments are quite satisfactory, specially if we think that we are training our system with little data. As expected, the FF algorithm reports a better accuracy.

4.3 Error analysis

In a brief error analysis we can state that:

- The factor that most penalizes the performance of the classifiers is the size of the corpus. This is particularly critical in the CINEMA corpus, since not only the number of utterances for training is low, but also the number of utterance per category can be very low (goes from 1 to 38), meaning that some categories only have a unique utterance associated. When this type of utterance appears in the test set of a classifier, failure is almost certain. We should also say that this situation did not occurred in the DOMOTIC corpus, because the number of utterances by category is very homogeneous.
- What also happens when we have a small corpus is that the system easily gives special weight to some words that do not have any particular meaning. For instance, consider a category where almost every utterance has the expression *How to...* and consider also that all other categories do not have this expression. In this scenario the classifier is likely to classify all utterances that have the words *How to...* with the same category. This problem can be attenuated with the use of the important word/expression list.
- Many times a certain verb conjugation or word synonym appears only in the test corpus and therefore the classifier fails. Thus, it would be useful to have some sort of normalizer (a stemmer, for instance).
- When the classifiers fails the returned score is low. Therefore, we can identify these situations and, in the case of the frames, explicitly ask the user for the values of the attributes which had a low confidence score.

5 Conclusions and Future Work

In this paper we have shown how natural language understanding can be implemented as a classification task, by treating in a uniform manner the process of understanding either the system should return an utterance, a logical form or a frame. Several features are implemented and a named entity recognizer is already incorporated. In this paper, we have described how the mapping of an utterance into an utterance, logical form and frame can be done within a classification process and we have tested our approach in three scenarios representing the different outputs. Moreover, we have dedicated particular attention to the use of lists as features, and we have explored the possibility of creating these lists in an automatic process. We have concluded that a list that is to be used as a feature obtains almost the same results either it has been built manually or automatically.

Another problem that draws our attention was the use of named entities in the understanding process. We have shown that accuracy can be increased if, in a pre-processing step, named entities are replaced by the appropriate tag.

Finally, we have run two experiments with the goal of testing the importance of previously identifying a frame type, before classifying its attributes, thus, putting apart the need of using heuristics in order to combine attributes' results. Results were much improved with this approach.

Concerning future work, other understanding techniques could be easily incorporated and combined. In addition, the resulting system should be tested with real utterances in the ART and DOMOTIC domains.

Acknowledgments

This work was supported by national funds through FCT Fundao para a Cincia e a Tecnologia, under project PEst-OE/EEI/LA0021/2011.

References

1. Alfred V. Aho and Margaret J. Corasick, 'Efficient string matching: an aid to bibliographic search', *Communications of the ACM*, **18**, 333–340, (June 1975).
2. James Allen, Nathanael Chambers, George Ferguson, Lucian Galescu, Hyuckchul Jung, Mary Swift, and William Taysom, 'Plow: a collaborative task learning agent', in *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*, pp. 1514–1519. AAAI Press, (2007).
3. James F. Allen, Bradford W. Miller, Eric K. Ringger, and Teresa Sikorski, 'A robust system for natural spoken dialogue', in *Association for computational linguistics*, pp. 62–70, (1996).
4. *The Core Language Engine*, ed., Hiyan Alshawi, Bradford Books, MIT Press, 1992.
5. Niels Ole Bernsen and Laila Dybkjær, 'Domain-Oriented Conversation with H.C. Andersen', in *Proc. of the Workshop on Affective Dialogue Systems, Kloster Irsee*, pp. 142–153. Springer, (2004).
6. Rahul Bhagat, A. Leuski, and Eduard Hovy, 'Shallow semantic parsing despite little training data', in *Proc. ACL/SIGPARSE 9th Int. Workshop on Parsing Technologies*, (2005).
7. Dan Bohus, Antoine Raux, Thomas K. Harris, Maxine Eskenazi, and Er I. Rudnicky, 'Olympus: an open-source framework for conversational spoken language interface research', in *HLT-NAACL 2007 workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technology*, (2007).
8. Chih-Chung Chang and Chih-Jen Lin, *LIBSVM: a library for support vector machines*, 2001.
9. Pilar de Almeida and Yokoi Shigeki, 'Interactive conversational character as a virtual tour guide to an online museum exhibition', in *ICCE '02: Proceedings of the International Conference on Computers in Education*, p. 215, Washington, DC, USA, (2002). IEEE Computer Society.

10. Stefan Kopp, Lars Gesellensetter, Nicole C. Krämer, and Ipke Wachsmuth, *A conversational agent as museum guide: design and evaluation of a real-world application*, 329–343, Springer-Verlag, London, UK, 2005.
11. Anton Leuski, Jarrell Pair, David R. Traum, Peter J. McNerney, Panayiotis P. Georgiou, and Ronakkumar Patel, 'How to talk to a hologram', in *IUI*, eds., Cécile Paris and Candace L. Sidner, pp. 360–362. ACM, (2006).
12. Anton Leuski, Ronakkumar Patel, David Traum, and Brandon Kennedy, 'Building effective question answering characters', in *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pp. 18–27. Association for Computational Linguistics, (2006).
13. Anton Leuski and David Traum, 'NPCEditor: A Tool for Building Question-Answering Characters', (2010).
14. Anton Leuski and David R. Traum, 'A statistical approach for text processing in virtual humans', in *Proceedings of the 26th Army Science Conference*, (2008).
15. Anton Leuski and David R. Traum, 'Practical language processing for virtual humans', in *IAAI*, (2010).
16. Catarina Moreira, Ana Cristina Mendes, Lusa Coheur, and Bruno Martins, 'Towards the rapid development of a natural language understanding module', in *IVA 2011, the 11th International Conference on Intelligent Virtual Agents*, (September 2011).
17. Gokhan Tur and Renato De Mori, *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, Wiley, 2011.
18. Ye-Yi Wang and Alex Acero, 'Spoken language understanding: An introduction to the statistical framework', *IEEE Signal Processing Magazine*, 16–31, (2005).
19. R.M. Kaplan e B.N. Webber W.A.Woods, *The Lunar Sciences Natural Language Information System: Final Report*, volume BBN Report 2378, Bolt Beranek and Newman Inc., Cambridge, Massachussets, 1972.
20. Joseph Weizenbaum, 'Eliza - a computer program for the study of natural language communication between man and machine', *Communications of the ACM*, **9**(1), 36–45, (1966).