Extracting Primary Objects and Spatial Relations from Sentences

Neha Baranwal, Avinash Kumar Singh and Suna Bensch
Department of Computing Science, Umeå University, Umeå, Sweden
{neha, avinash, suna}@cs.umu.se

Keywords: Natural Language Grounding, Spatial Relation Extraction, Hobb's Algorithm, Human-robot Interaction, NLTK, Google Speech, Stanford Parser.

Abstract: In verbal human-robot interaction natural language utterances have to be grounded in visual scenes by the robot. Visual language grounding is a challenging task that includes identifying a primary object among several objects, together with the object properties and spatial relations among the objects. In this paper we focus on extracting this information from sentences only. We propose two language modelling techniques, one uses regular expressions and the other one utilizes Euclidian distance. We compare these two proposed techniques with two other techniques that utilize tree structures, namely an extended Hobb's algorithm and an algorithm that utilizes a Stanford parse tree. A comparative analysis between all language modelling techniques shows that our proposed two approaches require less computational time than the tree-based approaches. All approaches perform good identifying the primary object and its property, but for spatial relation extraction the Stanford parse tree algorithm performs better than the other language modelling techniques. Time elapsed for the Stanford parse tree algorithm is higher than for the other techniques.

1 INTRODUCTION

If robots are to work with and for humans in everyday life, they have to be equipped with advanced language and verbal communication capabilities which visual language grounding is a crucial part of.

Visual language grounding is mapping verbal utterances to visual scenes and includes an analysis of the verbal utterance that involves identifying a primary object among several objects, identifying the properties of objects and their spatial relations.

Additional difficulties are that every human has their own demands on communication and their own ways of speaking. For example, a specific request or command can be verbalized and uttered in many different ways by different persons. To handle this problem, it is required that the coordination between human and robot is very effective and the interaction quality is high (Dautenhahn, 2007), (Bensch et al., 2017), (Baranwal, 2017). For example, to understand a command like *"Give me the red cup left of the book"*, a robot must be able to identify the primary object and its property (i.e. *"red cup"*) as well as the other object in the command (i.e. *"book"*) and the spatial relation between these two objects (i.e. *"left of"*). In this paper we propose two language modelling techniques, one is computational modelling based on regular expressions and another one is Euclidian distance based. In the first technique, the extraction algorithm is created with the help of regular expressions and a flat tree structure using NLTK (Bird and Loper, 2004) is generated, while the technique based on Euclidian distance is used for relation extraction between words and the identification of the primary object together with its property (PPO) is done with the help of rules applied after pos tagging.

These two proposed language modelling techniques are further compared with tree-structure based algorithms. We have considered four cases of sentences as follows:

1. Sentences containing a single object such as *"Give me the red cup"*.
2. Sentences containing two objects and one spatial relation in between these two objects such as *"Pick up a book which is left of the yellow bottle"*.
3. Sentences containing multiple objects and multiple spatial relations, where we distinguish the following cases:

3.1 Each object contains at least one spatial relation such as in *"Give me a cup which is left of book on the table"*.

3.2 One object contains more than one spatial relation

such as in *"Pick up an orange which is behind the book and right of red cup"*.

For all techniques Google speech is used for speech-to-text conversion. We have solved all the four test cases in this paper and achieved 98% accuracy for PPO for all techniques, but the tree structure and Euclidean distance based algorithm perform better for subject object relation extraction. The overall architecture is illustrated in Figure 1.

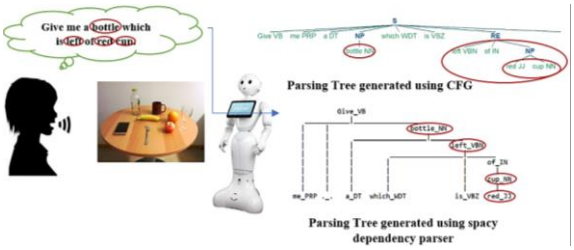

Figure 1: Illustration of the proposed overall architecture. Verbal utterances given by the human are analysed with our proposed language modelling techniques for further mapping to visual scenes.

The structure of the paper is as follows: Section 2 gives a literature overview of previous work on relation extraction and language grounding. In Section 3, all the four language modelling techniques are explained. Experimental results and analysis are discussed in Sections 4 and 5. In Section 6, the paper is concluded towards different language modelling techniques.

2 BACKGROUND

Most of the work has been done on name entity-based relation extraction or on language grounding. Golland et al., (2010) proposed a game theoretic model for language grounding where they tried to identify the spatial relation between objects. It is a "speak and tell" approach where a speaker generates an utterance containing an actual object and the listener tries to guess the object. If both objects are equal, then it is a success otherwise fail. They have evaluated their experiment with some constraints and achieve 78% accuracy. These constraints have been released in paper (Guadarrama et al., 2013) by using a probabilistic approach, in which a primary object and their spatial relationship with other objects is extracted from a visual scene. This information is combined with semantic parsing of sentences using template matching and a probabilistic approach. They achieve an accuracy about 84%. Olszewska (2017) built human/robot dialogues based on semantically meaningful instructions like the directional spatial relations represented by the clock model. Explainable AI is used for language grounding (Hendricks et al., 2018). In this method features extracted from visual and language module are provided to LSTM and apply 2-layer neural network to obtained the final score of grounding. In papers (Alomari et al., 2017; Alomari et al., 2016) grounding is performed using a robot control language (RCL) tree where visual learning is done with the help of color, shape and location feature obtained from an object. Direction and distance between pair of objects is extracted as a relation feature and finally the action performed by the robot is extracted from the video clip. These features are clustered based on their category and mapped with words using RCL tree. Preprocessing and word extraction from sentences is done using NLTK toolkit (Bird and Loper, 2004). In name entity-based approach they tried to extract the relation between person and organization, organization and city etc. Open relation extraction approach is proposed by (Banko and Etzioni, 2008; Banko et al., 2007) where lexico-syntactic patterns is used to build a relation independent model. Conditional random field (CRF) is used for classifying a relational token. Very less work is done on primary entity and their relation extraction with other objects with respect to our day to day life objects like knife, mango, bottle etc. In this paper we are proposing and analyzing different language modelling techniques extracting above mentioned components from sentences. This work will be extended to ground primary objects and their relations in visual scenes, paving the way for effective human-robot interaction where the human commands the robot that specifically acts as helper in our daily life.

3 LANGUAGE MODELLING TECHNIQUES

The proposed language modelling techniques consist of two modules, namely speech-to-text conversion (explained in Section 3.1 below) and the algorithmic part of language modelling (explained in Sections 3.2, 3.3, 3.4, 3.5 below). The algorithmic part of language modelling uses regular expressions, Euclidian distance, an extended Hobb's algorithm based on dependency parses and Stanford phrase structure parses. All four techniques have few common steps such as word tokenization, preprocessing, pos tagging and chunking but they differ in the extraction part.

3.1 Speech-to-Text-Conversion

Google speech engine (Honnibal, 2016; Google LLC, 2014) is used to record natural language instructions by a human to a Pepper robot. The speech signals are recorded at 16000 Hz frequency using the *pyaudio* function. The FLAC encoding is used for lossless conversion. All the utterances are recorded in a closed environment where speech-to-text conversion errors are negligible.

3.2 Language Modelling using Regular Expressions

The initial steps are word tokenization, pos tagging and chunking using the NLTK toolkit. To achieve the goal of PPO and spatial relation extraction, we design regular expression that are used to build syntactic subtrees. In Table 1, the left column contains labels of syntactic categories and the right column contains the regular expressions. We use Table 1 to build syntactic subtrees where the categories in the left column are the labels of the roots of the subtrees and the categories in the right column are the labels of the children. The category NP stands for noun phrase or adjective with noun, VP stands for verb phrase and RE/REL stands for Spatial/Positional Phrase. The input to our algorithm is a string *s* of the following form: $w_1(tg_1)$ $w_2(tg_2)$ ... $w_n(tg_n)$, where w_i is a word and tg_i is its tag. For example, *give*(VB) *me*(PRP) *leftmost*(JJS) *red*(JJ) *apple*(NN). We now scan the *n* tags of *s* and try to find a tag tg_i that occurs in a regular expression in the right column of Table 1. For example, given the sequence of tags VB PRP JJS JJ NN, we see that JJ and NN occur in the regular expression in the first row of Table 1. If a tag tg_i occurs in a regular expression, we construct a subtree *st* as follows: the category in the left column becomes the root label of the subtree *st*, and $w_i(tg_i)$ becomes its child label. For example, given *red*(JJ) *apple*(NN), the root of the subtree *st* is labelled by NP and its children are labelled *red*(JJ) and *bottle*(NN).

Then, using NLTK a flat tree *t* is constructed, where the root node is labelled by S and its direct children are labelled by the words and tags in the input string *s* and the root label of subtree *st* (which has children as explained above). A pictorial example is given in Figure 2.

Once the tree *t* is constructed, we start traversing tree *t* in breath first manner for PPO and spatial relation extraction.

Input: Input to the algorithm is a tree *t* (such as the one given in Figure 1 and 2).

Table 1: Syntactic categories and regular expressions for building syntactic subtrees. As usual, the symbol * denotes occurrence of any entity including zero, the symbol | denotes alternative, and the symbol ? denotes possible presence of an entity.

NP	<JJ>?<NN>|<NN*>+
RE	<VP>+|<CLAUSE>
REL	<VP><IN|TO|RP>|<IN|TO><NP>| <VP*><DT><NP>
VP	<VB.*><NP| REL |CLAUSE>+$
CLAUSE	<NP><VP>

Table 2: This table contains all words expressing spatial/positional information that are wrongly identified as noun phrases in the syntactic trees of the algorithm in Section 3.2.

Spatial/Positional Information
Left
Right
Among
Behind
Leftmost/ Extreme left
Rightmost/ Extreme Right
Middle/Center
In front of
Between
After
Within
On
In
Top
Bottom
Middle/Center Left
Middle/Center Right

1- Traverse tree *t* starting at the root node and identify all nodes labelled with NP.

2- If there is one single NP node *n* in *t*:

2.1 Traverse the branches of the subtree rooted at NP and identify all leaf nodes $n_1, n_2, ..., n_k$. Let the node n_k labelled with NN be the primary object. Let all other nodes $n_1, n_2, ..., n_{k-1}$, that are labelled with JJ be the properties of the primary object.

2.2 Traverse tree *t* starting at the root node and identify the spatial or positional information tagged with JJ or JJS given in Table 2.

3- If, in tree *t*, there are *k* NP nodes $np_1, np_2, ..., np_k$, $k \geq 2$, (as in Figure 3), then we store all labels of their children w_xJJ, w_yNN, etc. in an NP-list $[e_1, ..., e_k]$ as follows: all labels of all children of one NP-node np_i are one element e_i in the NP-list. For example, for two different NP nodes with children

red(JJ) cup(NN) and blue(JJ) mug(NN), respectively, we store as one element red(JJ) cup(NN) and blue(JJ) mug(NN) as another element in the NP-list.

Primary Object Identification:

3.1 Find all nodes *n* in the tree *t* labelled NP. Check whether elements e_i of the NP-list occur in Table 2 or not. If e_i occurs in Table 2, remove e_i from the NP-list and store it in a relation list called R. (Table 2 contains words that we want to eliminate because these words are spatial relation words which we do not want labelled NP).

3.2 After removing all e_i as explained above, the NP-list will contain the remaining words e_j. From this list choose the first e_1 and let it be primary object associated with its property (if present).

Relation Extraction:

3.3 Traverse the tree *t* and find a node n_r labelled RE or REL. Let the subtree rooted at RE or REL be *sr* (as given in Figure 3).

3.4 Traverse the subtree *sr* and find a node labelled with NP. Let *snp* be the subtree rooted at NP. If the leaf node in *snp* is not the primary object, then let the spatial relation represented in *sr* be the spatial relation of the object represented in the subtree *snp*. Furthermore, the object and spatial relation represented in *sr* are in relation to the primary object in tree *t*.

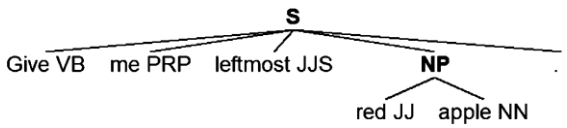

Figure 2: Parse tree generated using regular expressions for sentences having one single object.

3.3 String based Model using Euclidian Distance

As above the NLTK toolkit is used for word tokenization and POS tagging. This approach solves all the test cases with a single algorithm discussed below under relation extraction.

Input: Let the input be *s*, where *s* is of the form $w_1(tg_1)$ $w_2(tg_2)$ ... $w_n(tg_n)$, where w_i is a word and tg_i is a tag.

If Input *s* has only One Object (i.e. One NN Tag):

1- For w_i(NN) in s, check if w_i occurs in Table 2.
2- If w_i occurs in Table 2, place w_i into the relation list R.
3- Else w_i is considered as primary object.
4- Let w_i(NN) be the primary object in input *s*. If w_{i-1}(JJ), then w_{i-1} is the property of w_i.
5- Search the closest w_j(VB) or w_j(JJS) to w_i:
6- If w_j(VB) or w_j(JJS) occur in Table 2, consider w_j as the spatial relation to the primary object w_i.

If Input *s* has Multiple Objects (i.e. NN Tags):
Primary Object Identification:

7- Step 1 and 2 are same for multiple objects.
8- If w_i does not occur in Table 2, place w_i into an object list O and apply Step 4 for all w_i.
9- If w_{i-1} with tag JJ is not in Table 2, w_{i-1} is the property of the object w_i and we store w_{i-1}(JJ)w_i(NN) in a list *e* as one element, $e = [e_1 \ldots\ldots\ldots e_n]$, with $e_i = w_x$JJ w_yNN.
10- The first element of list *e* is the primary object with its property.

Relation Extraction:

11- Search all verbs w_i (tagged w_iVB, w_iVBD, w_iVBN, w_iVBZ, w_iVBG) in list *e*.
12- If w_i is in Table 2, place it into the relation list R.
13- Find out the distance between the elements of the relation list R and the object list *e* except the primary element:

$$d(R, e)=\sqrt{\sum_{i=1}^{n}(R_i - NN_i)^2}$$

14- Select the minimum distance relation treated as a relation of primary object with respect to another object.

3.4 Dependency Parser based Language Model (Reformulated Hobb's Algorithm)

In this approach language modelling is done after generating the parsing tree for a sentence. A parsing tree is generated using Spacy dependency parser (Spacy Inc.) (see Figures 3 and 4) where stop words (e.g. is, am, are, the, was, were, of) are removed using the NLTK toolkit.

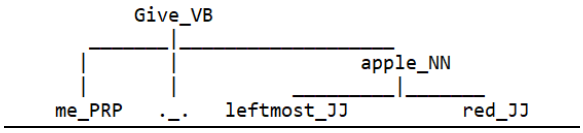

Figure 3: Parse tree generated using Spacy dependency parser for a sentence having a single object.

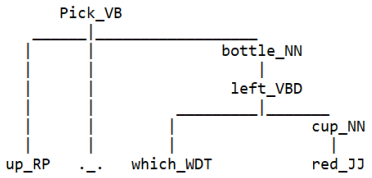

Figure 4: Parse tree generated using Spacy dependency parser for a sentence having two objects.

The Spacy dependency parser is a syntactic parser useful for sentence boundary detection. After parsing, a reformulated Hobb's algorithm is applied to extract the primary object, its property and spatial relations. In 1977, Hobbs (Hobbs, 1978; Lappin and Leass, 1994) proposed two pronominal anaphora resolution algorithms, where one is syntax based and the other one is semantic based. The syntax based approach is applied in this paper which is redeveloped to extract the primary object, its property and spatial relations with other objects from a sentence. This algorithm solves all the three cases simultaneously. The reformulated Hobb's algorithm is discussed below:

Input: Let *t* be an input dependency tree. All nodes are labelled by labels of the form *x*_CAT, where *x* is a word and CAT is a syntactic category (such as in the example displayed in Figures 3 and 4).

1- Traverse the tree *t* in breath first manner (left-right) and identify all words w_i having tag NN (w_i NN) and check whether w_i occurs in Table 2. If yes, store the word w_i into relation list R, else proceed.
2- If there is only one single node *n* labelled *x*_NN:
3.1 Let *n* be the primary node.
3.2 Traverse all the children of *n* from left to right and identify all nodes n_j labelled with *x*_JJ (representing adjectives).
3.3 If *x* in label *x*_JJ occurs in Table 2, store *x* in R.
3.4 Else *x* is the property of the NN node.
3- If there is more than one node labelled NN:

Primary Object Identification:

3.1 Begin with the lowest x_NN node in *t* and traverse all its branches and find all nodes labelled *x*_JJ and *x*_V, where V = {VB, VBD, VBN, VBZ, VBG}.
3.2 If *x* in label *x*_JJ or *x*_V occur in Table 2, store *x* into R.
3.3 Else *x* in label *x*_JJ is the property of *x* in *x*_NN.
3.4 Again, go up the tree and repeat Steps 3.1, 3.2 and 3.3.
3.5 This process continues until we hit the node labelled *x*_NN which is closest to the root. This *x* in *x*_NN node is called primary element and *x* in *x*_JJ is called its property.

Relation Extraction:

3.6 Begin with lowest *x*_NN node in *t* and go up the tree and find *x*_V, V = {VB, VBD, VBN, VBZ, VBG} or *x*_NN node.
3.7 If *x* in label *x*_NN or *x*_V occur in Table 2, then denote *x* as spatial relation (R).
3.8 This spatial relation (R) is a relation between the primary object and the object considered in Step 3.6
3.9 Repeat Step 3.6, 3.7 and 3.8 until the primary object is identified.

3.5 Stanford Parse Tree based Language Model

Input: Let *t* be an input parsing tree produced by the phrase structure Stanford parser (Marneffe et al., 2006). All inner nodes in *t* are labelled by tags and the leaf nodes of *t* are labelled by words (such as in the example displayed in Figures 5 and 6).

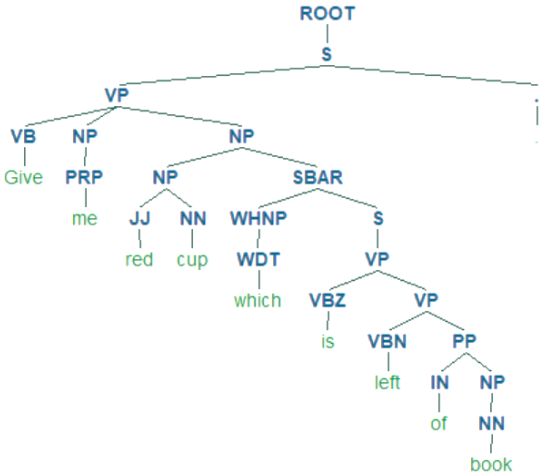

Figure 5: Parse tree generated using Stanford parser for a sentence having two objects.

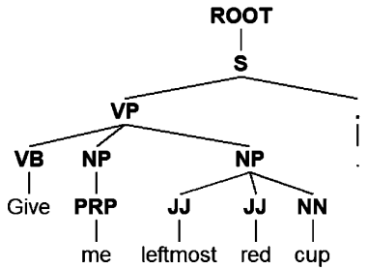

Figure 6: Parse tree generated using Stanford parser for a sentence having a single object.

1- Begin with the root node and traverse all branches of *t* in breath first manner and find all NP nodes.
2- If there is only one single NP node *n*:
2.1 Traverse the branches of the subtree rooted at NP and identify adjectives (JJ) and the name of objects (NN).
2.2 If the child *x* of JJ or NN occur in Table2 store *x* into the spatial relation list R.

2.3 Else child *x* of NN is the primary object and child *x* of JJ its property.

3- If there is more than one node labelled NP:

Primary Object Identification:

3.1 Start with lowest NP node in *t* and traverse its children. If there is an NN node, store it into the object list O together with its property (if present); the steps are same as in 2.1, 2.2 and 2.3.

3.2 Traverse up the tree and repeat Step 3.1.

3.3 The object list contains all the objects associated with its property in last in first out manner.

3.4 The last object is called primary object.

Relation Extraction:

a) Sentence Having Two Objects:

3.5 Begin with lowest NP node in *t*, go up the tree *t* and find the VP or PP node.

3.6 If there is a VP or PP node, traverse left children below VP and PP node.

3.7 If the children are in Table 2, store them in the relation list (R).

3.8 This spatial relation (R) is a relation between the primary object and the last NP node discussed in Step 3.5.

3.9 Go up the tree and repeat Steps 3.5, 3.6, 3.7 and 3.8.

b) Sentence Having More than Two Objects:

Case1: One Object Is Associated with One Spatial Relation

3.10 Begin with lowest NP node in *t*, go up the tree and find the VP, PP or NP node.

3.11 If VP, PP or NP node is present, traverse left children.

3.12 If the children are in Table 2, store them in the relation list (R).

3.13 This spatial relation (R) is a relation between the primary object and the last NP node defined in 3.10.

3.14 Go up the tree and repeat Steps 3.10, 3.11, 3.12 and 3.13 until the primary object is identified.

Case 2: One Object has more than One Spatial Relation

3.15 All the steps are same as Case 1 except Step 3.11.

3.16 Step 3.11 is replaced with traversing all the children of VP, PP or NP nodes in breath first manner.

4 EXPERIMENTAL SETUP

We have considered the COCO dataset (Tsung-Yi et al., 2014) which contains objects eligible for a table top environment. Experiments have been performed on 200 sentences where 40 sentences had a single object and 160 sentences had 2 or more than 2 objects. 10 subjects have been considered for dataset collection. Primary object with their property and relation was extracted manually for validation purpose. We have also performed our experiment on strands dataset (Alomari and Dukes, 2016). This is a language grounding dataset having two parts one containing images and other containing sentences describing actions. We have considered 200 sentences from the language grounding dataset.

Parsing trees were generated using regular expression with the NLTK toolkit, Spacy dependency parser and Stanford parser, for all three categories of sentences (single object, two objects with single relation and multiple objects with multiple relations) as shown in Figures 2-6. The process flow of breath first approach is shown in Figure 7.

5 RESULTS AND DISCUSSION

Out of in total 200, 196 sentences have correct extraction of the primary object with their properties but only 171 sentences have correct extraction of relation of primary object with another object in case of language modelling using regular expression. It fails to predict the relations like *"middle left"* or very complex sentences like *"Could you possibly help me and get me my black phone kept on the extreme left bottom of the table".* In this sentence the modelling technique extracts *"left bottom"* but is unable to extract term *"extreme"*. The extended Hobb's algorithm successfully identifies these kinds of relations. Out of 200 sentences, 175 sentences have correct extraction of relation. This method fails in case of sentences like *"May I have that mug? That red one"* because we get two parsing trees using Spacy one for *"May I have that mug"* and another for *"That red one"*. The Euclidian-distance based algorithm fails to identify relations having the same distance with two objects. Among all the Stanford parser based algorithm performs better. On the strand sentence dataset (Alomari and Dukes, 2016) all the methods provide 97.2% accuracy on PPO and 93.1% with CCG, 94.4% with Euclidian, 94% with extended Hobb's and 95.3% with Stanford in case of relation extraction.

Comparative analysis between all the modelling techniques on in house dataset is shown in Figure 8 and the time elapsed by all four algorithms is shown in Table 3.

After analyzing Table 3, we observe that tree traversing takes longer time than the regular-expression

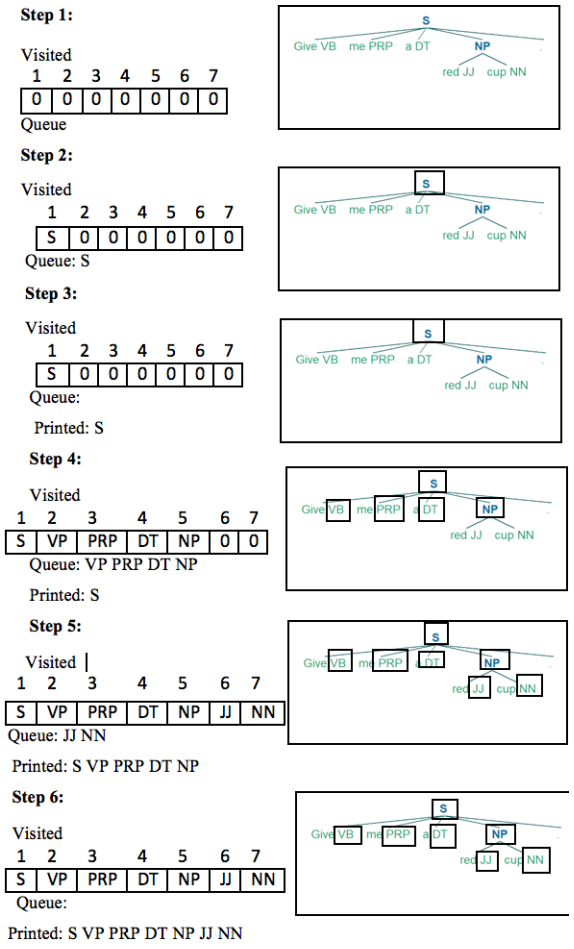

Figure 7: Process flow of language modelling using regular expressions.

based approach and string based approach (Euclidian distance).

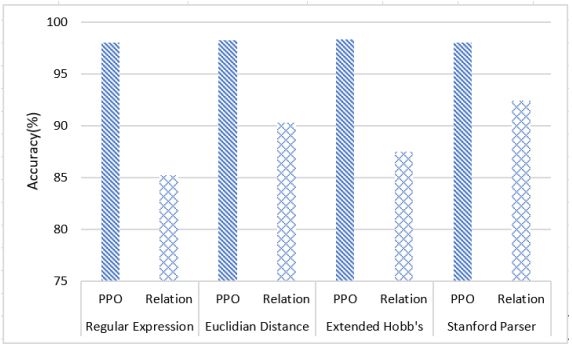

Figure 8: Comparative analysis between all language modelling techniques investigated in this paper.

Table 3: Time elapsed by all four algorithms considered in this paper.

Method	Time Elapsed (Sec.)
Regular Expression	0.00983
Euclidian Distance	0.00968
Extended Hobb's Algorithm	0.10251
Stanford parsing	0.91921

6 CONCLUSION

Four approaches for language modelling were performed where we proposed two new approaches. One is a string based approach (Euclidian distance) and another is regular expression based. These two approaches are further compared with two other tree traversing based approaches. All the methods work good for the extraction of primary object and their property but the Stanford parser based algorithm works better than others in case of relation extraction. In such cases where the sentences having clear instructions of object to object relation, the regular expression based algorithm performs good but fails to identify if the sentences are complex. The best result was achieved with the Stanford parser based algorithm, giving an accuracy of 98% for primary object with property, and 92.4% for relation. The time elapsed with the Stanford parser based method is higher than others. Among all we observe that Euclidian distance based method performs good with minimum CPU time. This work is planned to be further extended with anaphora resolution to deal with multi sentences such as "*The close yellow bowl, the cup next to it and the green bowl next to it"*.

ACKNOWLEDGEMENTS

This research was made possible by the research funding from the Swedish Kempe foundation for Neha Baranwal and Avinash Kumar Singh.

REFERENCES

Alomari, M., Duckworth P., Hawasly, M., Hogg D. C., Cohn, A. G., 2017. Natural language grounding and grammar induction for robotic manipulation commands. *In Proc. of the First Workshop on Language Grounding for Robotics*, pp. 35-43. ACL.

Alomari, M., Duckworth P., Gatsoulis Y., Hogg D.C., and A.G. Cohn, 2016. Unsupervised Natural Language Acquisition and Grounding to Visual Representations for Robotic Systems. Workshop on Cognitive

Knowledge Acquisition and Applications (Cognitum 2016), IJCAI 2016.

Alomari, M., Dukes, K., 2016. *Extended train robots. Dataset.* Uni. of Leeds. https://doi.org/10.5518/32

Banko, M., Etzioni, O., 2018.The tradeoffs between open and traditional relation extraction. *In Proc. of ACL-08: HLT*, pp. 28–36. ACL.

Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O., 2007. Open information extraction from the web. In *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence*, pp. 2670–2676.

Baranwal, N., 2017. Development of a Framework for Human-Robot interactions with Indian Sign Language Using Possibility theory. *Int. J. of Social Robotics, 9: 563-574*. Springer.

Bensch, S., Jevtić A., Hellström T., 2017. Interaction Quality in Human-Robot Interaction. *In Proc. of the 9th Int. Conf. on Agents and Artificial Intelligence (ICAART) 2017, Volume 1*, pp. 182-189.

Bird, S., Loper E., 2004. NLTK: the natural language toolkit. *In Proc. of the ACL 2004 on Interactive poster and demonstration session*, p. 31. ACL.

Dautenhahn, K., 2007. Socially intelligent robots: dimensions of human–robot interaction. *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 362, no. 1480: 679-704.

Golland, D., Percy L., Klein, D., 2010. A game-theoretic approach to generating spatial descriptions. *In Proc. of the 2010 conf. on empirical methods in natural language processing*, pp. 410-419. ACL.

Google LLC, 2014. https://cloud.google.com/speech-to-text/

Guadarrama, S., Riano, L., Golland, D., Gouhring, D., Jia, Y., Klein, D., Abbeel, P., Darrell, T., 2013. Grounding spatial relations for human-robot interaction. *In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ Inter. Conf.* pp. 1640-1647. IEEE.

Hendricks, L. A., Ronghang H., Trevor D., Akata, Z., 2018. Grounding visual explanations. *arXiv preprint arXiv:1807.09685*.

Hobbs, J. R. 1978. Resolving pronoun references. *Lingua 44*, no. 4: 311-338.

Honnibal, M., 2016. "Introducing spaCy". explosion.ai. https://spacy.io/api/dependencyparser.

Lappin, S., Leass, H.J., 1994. An algorithm for pronominal anaphora resolution. *Comp. Ling. 20, no. 4: 535-561*.

Marneffe, D., Catherine, M., MacCartney, B. Manning C., 2006. Generating typed dependency parses from phrase structure parses. *Proc. of LREC. Vol. 6. No. 2006*.

Olszewska, J. I., 2017. Clock-model-assisted agent's spatial navigation. *Proc. of the 9th Int. Conf. on Agents and Artificial Intelligence (ICAART), Volume 2, 687--692*.

Tsung-Yi, L., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context. *European conference on computer vision*, pp. 740-755. Springer.