A Further Step Towards Automatic Domain Modelling by Relevant Information Extraction

Lars Krupp
Embedded Intelligence
DKFI Kaiserslautern
Kaiserslautern, Germany
n.n@dfki.de

Gernot Bahle
Embedded Intelligence
DKFI Kaiserslautern
Kaiserslautern, Germany
n.n@dfki.de

Agnes Gruenerbl
Embedded Intelligence
DKFI Kaiserslautern
Kaiserslautern, Germany
n.n@dfki.de

Paul Lukowicz
Embedded Intelligence
DFKI and TU Kaiserslautern
Kaiserslautern, Germany
n.n@dfki.de

***Abstract*—A still unsolved issue in human activity recognition, as it is being used in many smart systems, is the availability of labeled training data or, in other words, information about what sensors actually are recording. A possible solution to this problem is to build detailed semantic domain models specifying, in different detail, complex compound activities. Such models would allow retrieving the required information without the necessity of time-consuming labeling. On our way to develop a method to leverage text-based domain descriptions automatically to build such domain models, we previously introduced a method to extract domain-relevant information from texts. However, this method still included the requirement to hand-craft so-called "regular expressions," which have to be adjusted or rebuilt for different languages and different styles. Even though only done once for a language or style, the necessity of hand-crafting regular expressions (and their fine-tuning by experts) still requires extensive work. In this paper, we present the next step to automatize information extraction by substituting the regular expressions with an automated neural network. All steps in this method are now fully automatic and do not require any hand-crafting. The performance of this new method is equal to the performance of the regular expressions method before (70% precision and recall).**

***Index Terms*—automatic information extraction; complex activity recognition; domain models**

I. Introduction

Recognition of human activities from wearable and ubiquitous sensor data is an active research field (e.g. Wang et al. [1]). Simple recognition tasks such as step recognition or modes of locomotion (e.g. driving, walking, etc.) are already well integrated in commercial platforms (e.g. fitness trackers and smart phones [2]).

More complex recognition tasks, however, still present challenges. One of the major problems in this context is the acquisition of the necessary label data for supervised training schemes. In cases of only a couple of people doing only a limited set of activities, gathering label-data from video footage by hand is a possibility.

Nevertheless, even in that case, the amount of manual labor is intensive. As soon as the data-set should include a large amount of persons, more complex activities, or several subjects at once, the effort to acquire ground-truth becomes extensive or even prohibitive. Another option, i.e. asking test subjects to document all of their activities themselves, is not ideal either. Subjects would need to interrupt their activities very frequently to provide timely labels, or, when done after the sequence of activities was finished, rely on their memory, which may be prone to inaccuracies and errors.

Thus, in an ideal world, it would help a lot if labels could be retrieved automatically. As a precondition, the fact that the workflow of certain tasks is often known and only allows a certain variance can be leveraged. As an example, when baking a cake, a batter has to be prepared before the cake could go into the oven; for the batter, liquid ingredients have to be mixed before dry ingredients can go in; the order of the liquid and dry ingredients within their respective step, however, can be freely chosen.

Out of this knowledge of what can or cannot happen, recognition schemes to determine complex activities can be designed that do not operate in a classical supervised way. Essentially, this happens by decomposing complex activities into basic actions for which reliable recognition systems are either available or can be trained easily, then building compound complex activities out of them using logical and other constraints (e.g. order of execution, required items, location, etc.). This approach has been suggested [3].

A precondition for such an approach are (semantic) models that describe (to some detail) how the complex activities are composed. Such models would then provide the required information about what was supposed to happen at which step of a task. Clearly for some dedicated or small applications such models can be hand-crafted (e.g. [4]). However, for the approach to scale to a broader range of activities and applications, ways of automatic generation of such domain models are needed.

In this paper we built on previous work [5] where we introduced a method to extract semantic information from online domain descriptions about the way complex activities (for a specific domain) are composed of simple ones that have to be performed (e.g. a manual for assembling a furniture piece). This extracted relevant information was then used to automatically compose parts of tree-based semantic models of different domains.

However, even though the extraction of domain-relevant information provided decent results (precision of 88% and recall of 77%), the major drawback of this method was the requirement to build so called (partially quite complex) “regular expressions” per hand for different languages and also for different complexities within a language. Despite the necessity of composing these regular expressions per hand however, our previous work has the advantage that we are able to extract full sentences and not only information excerpts, as done in other work! This is important since, as outlined in our previous work, the composition of a sentence often carries information about the structure of a task. For example: First the liquid ingredients have to be mixed, then the dry ingredients can be added to the batter. If we would only extract to “ add the liquid ingredients” and to “add the dry ingredients” the batter would not have the required consistency. Therefore, we wanted to be able to extract not only information but also those sentences containing the required information with their structure.

Here, in this paper we now present a method, based on neural networks, that keeps the positive aspects of our previous work, but takes over the work of the “regular expressions”, which allows to automatize all steps in the process of extracting all relevant information about a domain within a written text. This provides another step towards fully automatically composition of domain models out of text-descriptions. Them main contribution and novelty of this paper is the automation of a step that previously had to be done manually, which makes the proposed approach more useful and also independent of complexity-levels in the language. In the previous method the regular expressions had to be adjusted according to the complexity-level of the used language, which is not necessary when using neural networks. Furthermore, we doubled the data-set used to evaluate the methodology; it now includes 21 (previous work included 11) domain descriptions with an average of about 41 sentences, resulting in a data-set of 872 sentences (of which 464 actually contain relevant information about their respective domain) to be tested.

II. Related Work

For extracting information (IE) a variety of methods exist. On the one hand, there is domain specific machine-learning to extract the necessary rules from annotated data, as in [6] and [7] for example. This method allows for high adaptability to domains, but requires the availability of enough annotated data to extract adequate rules. This condition makes these algorithms not usable for our needs. The rule-based approach on the other hand, as in [8] for example, requires an expert to manually constructs specific rules. This, furthermore, is a complex process that requires an adequate knowledge base regarding the domain in which the IE-system is supposed to work.

Another approach for information extraction is open IE, like [9], which uses self-supervised learning. This approach provides automatic labeling of data using a parser, which leads to domain independence, but requires a language-specific classifier. Despite its positive aspects (e.g. domain independence), open IE [10] performs worse than domain dependent IE (e.g. [6]), particularly in the recall. Therefore, open IE methods do not suffice for our purpose.

Many of these methods make use of word embeddings [11] which are normally used to quantify the semantic similarity between two words within a certain vocabulary. However using word embeddings for part-of-speech (POS) tag classification seems to be something that is normally not done since POS tags and their respective words are viewed as two different things.

The attempt to extract relevant information from text in practice is not new. Already 20 years ago [12] and [13] introduced a method to extract relevant keywords of biological information directly from scientific literature. These were selected by their relative accumulation in comparison to a domain-specific background distribution. However, this work is very specific and very domain dependent. Furthermore, in contrast to our requirements, this method only looks for specific keywords and not entire sentences.

Work by Patwardhan et al. [14] described an information extraction system based on a relevant sentence classifier to identify relevant regions and extract domain specific patterns. More recent work by Salloum et al. [15] present a clause-based framework for information extraction in textual documents, specifically focusing on relation extraction. In another domain Vivani et al. [16] propose an ontology-driven approach to identify events and their attributes from episodes of care included in medical reports written in Italy.

All work above only extracts specific parts of information and not information of an entire domain, as required for our goals, thus none of these methods were suitable for our needs.

III. Methodology

In our initial attempts to provide automatic extraction of relevant information from text [5] we used a part-of-speech tagger on a pre-processed document (see previous paper for details about the pre-processing steps) to enable the use of chunking (using regular expressions in the POS labeled part of the tuples to determine whether the sequence of tuples is relevant). For this process, the regular expressions had to be hand-crafted for each language and also for different complexities within a language. These expressions were designed to extract those sentences whose syntax indicated that it includes an action (e.g. “Verb Object”, i.e. “do something”). The requirement to hand-craft these regular expression, even though only once for a desired language standard, was a major drawback. In the following, we describe how we managed to exchange these hand-crafted regular expressions with a fully automatic Artificial Neural Network (ANN).

Instead of using hand-crafted regular expressions of the syntax of sentences, for this work we trained an Artificial Neural Network (ANN) to recognize, based on the syntax of a sentence, if it does or does not contains an action. If an action was found, the sentence was deemed relevant for the purpose of building a semantic model of compound activities, since these consist of hierarchically connected basic actions.

This was done by first transforming labelled and indexed sentences into their syntactical equivalent, as seen in 1, using the part-of-speech tagger provided by the NLTK [17]. This mechanism of pre-possessing has already been employed in the previous work referenced in [5], so details can be found there. Breaking down a sentence into syntactic specifiers for each word allows us to construct a complete vocabulary that is independent of the content of the text, while retaining the internal structure of the sentences to be classified.

Attaining such a vocabulary is not achievable when considering the entirety of the English language in respect to every domain imaginable. This is one of the reasons why domain-independent information extraction methods normally perform worse in comparison to domain-dependent methods. As previously shown, it is possible to predict, with a reasonably high success rate, if a sentence includes an action or not, solely by inspecting the part-of-speech tags of the sentences in question [5]. However, in contrast to the complete English dictionary, including all technical terms and domain specific words and contexts, the number of different part-of-speech tags is finite.

On the basis of the “Penn Treebank tag set” (see [17]) our vocabulary was created by adding symbols to it which are relevant for the meaning of a sentence. For example the comma and the closed bracket are considered to be possible words. There is also one word reserved for unexpected symbols, which is necessary to guarantee that the network can compute any kind of text, even if it includes non-ASCII symbols.

733 0 Have you always wanted to impress people with your art?	733 0 VBP PRP RB VBD TO VB NNS IN PRP$ NN .

Fig. 1. The right part shows how all words in a sentence (left) are broken down into semantic specifier tags. For example ”wanted” is assigned the tag ”VBD” which means verb, past tense.

With this method we were able to reduce the vocabulary to less then 50 different “words”. This drastically reduces computation time and allows for the resulting network to be domain-independent. There is also a smaller data set required to adequately train the network since the possible number of “word” combinations is smaller. This, in a way, can be seen as a feature reduction used to reduce the “curse of dimensionality” [18].

To train the ANN a new data set with 872 labeled sentences was used. Punctuation marks counted as words, as they are able to alter the meaning of a sentence a great deal. The number of words in a sentence ranged from 2 to 134 (average number of words in sentence is 68, including punctuation).

The data set is comprised of 21 “how to” domain descriptions. Please note that “how tos” were taken from various and quite differing practical domains, since our proposed method should be domain independent, i.e. work an any kind of different activities:

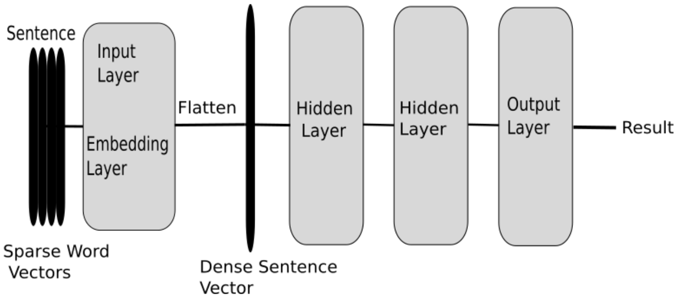

Fig. 2. Neural Network Structure: Shows the structure of the network and how the sentences are transformed from a matrix to one vector.

- changing a tires
- writing a summary
- building a lamp
- carving a pumpkin
- lighting a fire
- traveling to Europe
- planting a tree
- starting a business
- building a table
- identifying spiders
- growing bonsai trees
- improving writing speed
- finding a hobby
- building a power supply
- setting up a commercial kitchen
- training a pig
- caring for triops
- improving reaction speed
- jumping from high places
- going trekking in the Himalayas
- what to do in an emergency (A2E Method)

As evident, this data set contains a multitude of different domains and since it is written by a plethora of different people with different consumers in mind, it also includes multiple different writing styles. It is an expansion of the data set that was constructed for the training of the regular expressions before in [5].

All of those sentences were compiled into a single document such that each sentence occupied exactly one line in it. Every sentence was then indexed and labeled, either as 1 for relevant or as 0 for irrelevant sentences. Before training, the sentences were shuffled. This randomized the input document before each cross validation iteration, meaning sentences did not come in order of their domain, but random.

After this, a part-of-speech tagger was used to tag the data set. From this a new data set was created that consists of index, label and a sentence of POS-tags and symbols. Note, the automatic pre-processing steps were basically the same as in our previous method, but while our previous evaluation treated each domain individually, this time all sentences of all domains were put into "one pot" - as one data-set with many sentences. This combined set of sentences was then fed to the ANN in order to train it.

The implementation for this particular fully connected neural network was done using Keras [19]. As a first step, each of the sentences described above was padded such that all sentences would contain the same number of words. Then all words were one-hot encoded. The one-hot encoded sentence matrices resulting from this were then used as input for the word embedding layer.

The output, a non-sparse matrix representing the word-embeddings of a sentence, was then flattened and used by the artificial neural network to train. This was done until the network could decide if a sentence contains relevant information, or not. Relevant, in our case, as mentioned above, are sentences that describe an action. Afterwards for all inputs that were deemed relevant, the initial sentence is looked up and added to a list.
The results are a list of relevant sentences that were extracted from the initial text. Figure 3 shows the entire process.

The Artificial Neural Network(ANN) structure itself is fully connected network consisting of a word embedding layer which is also its input layer followed by two dense hidden layers and an output layer as seen in Figure 2.

In stark contrast to the previous method using hand-crafted regular expressions, this method works completely autonomous once the structure of the neural network is created. No extensive optimization by an expert is necessary.

To give a practical example, we use a sentence out of the sub-chapter "disabilities" from the ABCDE-method [20]. This example sentence is: "Limb movements should be inspected to evaluate potential signs of lateralization." In the pre-prosessing step it is automatically transformed into single words and signs, according their types. Please refer to [5] for details about this step. Now it looks like this:

('Limb', 'NN'), ('movements', 'NNS'), ('should', 'MD'), ('be', 'VB'), ('inspected', 'VBN'), ('to', 'TO'), ('evaluate', 'VB'), ('potential', 'JJ'), ('signs', 'NNS'), ('of', 'IN'), ('lateralization', 'NN'), ('.', '.')

Now the POS-tags are taken and transformed into integers by the vocabulary, which in our example would be:

12, 13, 11, 30, 33, 28, 30, 7, 13, 6, 12, 40

Then the sentence is padded with zeros until it has the right length to be put into the ANN. Here it is evaluated and a label is returned, which is either 1 or 0 depending on the relevance of the sentence. Now, if the sentence was deemed relevant, its index is used to obtain the original human-readable sentence. This is then added to the output list.

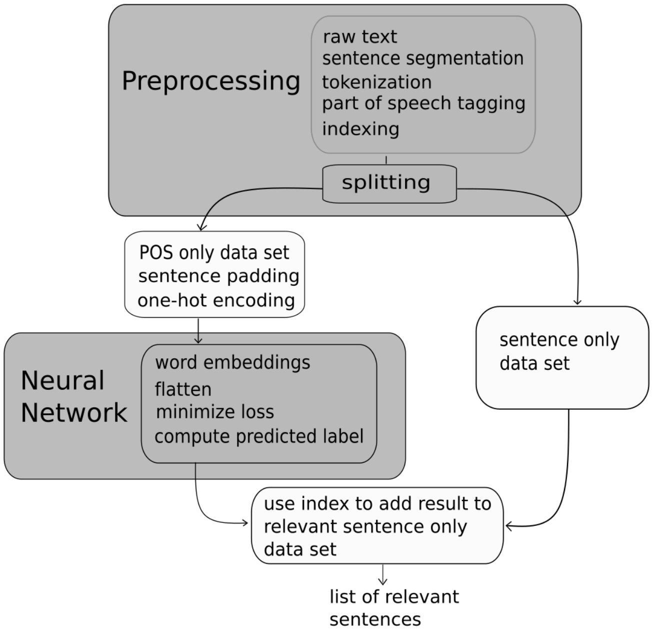

Fig. 3. All steps in the work-flow of our method are fully automated. In comparison, our initial work required the construction of a number of regular expression and their adaptation (mostly for every domain, since they were written by different persons and in different styles) instead of the fully automatic Neural Network. The automatic pre-prosessing part is the same for both methods.

IV. Evaluation

To evaluate the neural network we used a 10 times 10 fold cross-validation with different numbers of training iterations (these are also called "epochs" in the following sections). We also looked both at the data set as a whole and at subsets containing only sentences of different word lengths. For each of these cases, we calculated average precision and recall as evaluation metrics.

Since the data set has been doubled compared to our previous work using regular expressions, we also applied the old technique using regular expressions to the new data set.

This yielded a recall of 76% and a precision of 60%. In essence, this implies that while the regular expressions technique is capable of retrieving 3 out of 4 of all relevant sentences, the correctness of the retrieved samples is only 10% above random chance. Please note that compared to earlier work, this deterioration is due to trying to impose the regular expressions on a significantly more diverse data set than in the previous iteration. In other words, the regular expression approach does not scale well without adding more expert knowledge and would need significantly higher tuning to cope.

The results of the ANN show equal recall and precision of about 70%. While recall is slightly lower (by 5%) than the regular expressions approach, precision is noticeably higher (by 10%). Additionally, the new approach eliminates the major drawback of our previous iteration, namely the requirement to hand-craft the regular expressions. Here, all relevant steps can be performed automatically.

A. Training Time

The amount of training iterations used for neural networks is often an important characteristic determining both the computational effort necessary for setting up a trained model as well as the quality of the results they achieve at their tasks. To understand the behaviour for different numbers of iterations, we have tested our data set for 50, 100, 150 and 250 epochs. Table I lists the results. Unsurprisingly, training time increases with training iterations, though in a sub-linear way. However, since recall and precision remain the same for all epoch values, there does not seem a point to go higher than 50.

Epochs	Train. Time in Sec	Precision	Recall
50	827.7	70.3 %	70.1 %
100	1335.5	70.2 %	69.9 %
150	1930.1	69.9 %	69.9 %
250	2879.9	69.7 %	69.4 %
reg. expressions		59.5 %	75.8 %

TABLE I
INCREASING TRAINING TIME DOES NOT CHANGE THE OUTCOME. THEREFORE, ONLY A SMALL NUMBER OF TRAINING ITERATIONS IS REQUIRED FOR OUR PROPOSED METHOD.

B. Sentence Length

Another aspect that might have an influence on the performance of the ANN is the length, i.e. the word count, of sentences. There were both very short and very long sentences in addition to a majority of medium length ones. In order to evaluate the effect of different sentence lengths', we extracted all short sentences (5-10 words) from the data-set and evaluated the ANN with only short sentences, and the same with average length sentences (11-24 words) and longer sentences (25-50 words). Beyond 50 Words there were only 3 sentences with an extreme of up to 134 words, which were included in the over all data set, but not specifically tested in terms of sentence length.

Table II shows that the performance of the ANN for short sentences is noticeably better than the overall performance (10% better). Recall for medium and long sentences is about equal to the complete data-set (loosing about 2.5%). Precision for medium length sentences shows similar behaviour to the complete data-set (also loosing about 2.5%), but drops noticeably for longer sentences (by 10%).

Overall, sentence length seems to negatively impact ANN performance, though recall seems quite stable, while precision decreases with increasing word count.

V. Discussion and Conclusion

We have presented a way to extract meaningful sentences from textual representations of a variety of task descriptions in a fully automatic way using an ANN. These sentences can then be used to construct tree based hierarchical representations of semantic activities, alleviating the problem of ground truth scarcity in complex activity recognition. Compared to previous work in this domain, we achieve comparable results with significantly less expert input required.

	# of words	# sentences	Precision	Recall
short	5 – 10	153	79.4 %	81.2 %
medium	11 - 24	576	67.6 %	67.6 %
long	25 – 50	141	59.3 %	67.1 %

TABLE II
COMPARISON OF PERFORMANCE WITH DIFFERENT SENTENCE LENGTH. AVERAGE OF 500 TIMES 10-FOLD CROSS-VALIDATION

While 70% recall and precision seem a solid foundation, we believe further work can tune the neural network to achieve even better performance. For example, adding more layers with different functions or structure can be explored. Furthermore, adding different functions (e.g. for the distance calculations used in the flattening phase) may also improve results.

A. Outlook

An approach to leverage domain models and how they might look like - in this particular case hierarchical semantic tree models - has been introduces by [4]. As has been pointed out though, these kind of models had to be hand crafted, since mechanisms to turn text based descriptions of domains into adequate models have been missing.

Automatically extracting relevant information from text, as provided in this paper, obviously is the first step towards automatically generating such domain models. The next step is to turn the extracted information into the required model-shape. An approach to do so, has been introduced in our previous work Krupp et al. [5]:

After extracting relevant sentences it is possible to construct semantic trees as proposed by [4]. Such a tree one has to consider different possible logic operators like ”and” or ”if” and their relative position within a sentence. Since the English language is written and read from left to right this is in most cases the order in which the operators are dealt with. Furthermore the sentence is divided into multiple parts dictated by said operators. An ”and” for example would divide the sentence in at least two parts, the part before and the part after this operator. However the first part may be further divided in the case of an enumeration being detected. After this the operator acts as a parent node with the divided sentence parts as his children. These children are then checked for operators continuing the cycle until no operators are found. Applying this method results in a semantic tree which represents the meaning of the sentence.

Essentially, according to this method by Krupp et al., relevant sentences can be turned into single sub-trees providing the required action information in tree-branches. These sentence-sub-trees of a domain can then be arranged in a complete domain tree, as required in [4] for example. Once, such logically structured tree-based models are composed, they can assist activity recognition, because for each recognized action, the tree-model can provide information about the next most likely action or about the probabilities of different possible next actions. This information in turn should help any kind of activity recognition classifier to improve.

Acknowledgment

This work was supported by the HeadSense BMBF Project and AI4EU.

References

[1] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, 2018.

[2] G. Milette and A. Stroud, *Professional Android sensor programming*. John Wiley & Sons, 2012.

[3] M. Al-Naser, H. Ohashi, et al, and A. Dengel, "Hierarchical model for zero-shot activity recognition using wearable sensors." in *ICAART (2)*, 2018, pp. 478–485.

[4] A. Gruenerbl, G. Bahle, and P. Lukowicz, "Detecting spontan. collab. in dyn. group activities from noisy indiv. activity data," in *Pervasive Computing and Communications Workshops (PerCom Workshops)*.

[5] L. Krupp, A. Gruenerbl, G. Bahle, and P. Lukowicz, "Towards automatic semantic models by extraction of relevant information from online text," in *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 2019, pp. 481–483.

[6] D. Ciravegna *et al.*, "Adaptive information extraction from text by rule induction and generalisation," 2001.

[7] S. Soderland, "Learning information extraction rules for semi-structured and free text," *Machine learning*, vol. 34, no. 1-3, pp. 233–272, 1999.

[8] H. Cunningham, "Gate, a general architecture for text engineering," *Computers and the Humanities*, vol. 36, no. 2, pp. 223–254, 2002.

[9] M. Banko, M. J. Cafarella, et al., and O. Etzioni, "Open information extraction from the web." in *IJCAI*, vol. 7, 2007, pp. 2670–2676.

[10] L. Del Corro and R. Gemulla, "Clausie: clause-based open information extraction," in *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013, pp. 355–366.

[11] X. Ye, H. Shen, X. Ma, R. Bunescu, and C. Liu, "From word embeddings to document similarities for improved information retrieval in software engineering," in *Proceedings of the 38th international conference on software engineering*. ACM, 2016, pp. 404–415.

[12] M. A. Andrade and A. Valencia, "Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families." *Bioinformatics*, vol. 14, no. 7, pp. 600–607, 08 1998.

[13] R. Gaizauskas, G. Demetriou, P. J. Artymiuk, and P. Willett, "Protein Structures and Information Extraction from Biological Texts: The PASTA System," *Bioinformatics*, vol. 19, no. 1, pp. 135–143, 01 2003.

[14] S. Patwardhan and E. Riloff, "Effective information extraction with semantic affinity patterns and relevant regions," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 717–727. [Online]. Available: https://www.aclweb.org/anthology/D07-1075

[15] S. A. Salloum, M. Al-Emran, A. A. Monem, and K. Shaalan, "Using text mining techniques for extracting information from research articles," in *Intelligent natural language processing: Trends and Applications*. Springer, 2018, pp. 373–397.

[16] N. Viani, C. Larizza, V. Tibollo, C. Napolitano, S. G. Priori, R. Bellazzi, and L. Sacchi, "Information extraction from italian medical reports: An ontology-driven approach," *International journal of medical informatics*, vol. 111, pp. 140–148, 2018.

[17] E. Loper and S. Bird, "Nltk: the natural language toolkit," *arXiv preprint cs/0205028*, 2002.

[18] D. B. Dasari and V. G. Rao, "Text categorization and machine learning methods: current state of the art," *Global Journal of Computer Science and Technology*, vol. 12, no. 11, pp. 37–46, 2012.

[19] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[20] T. Thim, N. Krarup, E. L. Grove, C. V. Rohde, and B. Løfgren, "Initial assessment and treatment with the airway, breathing, circulation, disability, exposure (abcde) approach," *Int J Gen Med*, vol. 5.