Semantic role labeling tools for biomedical question answering: a study of selected tools on the BioASQ datasets

Fabian Eckert and Mariana Neves *
Hasso Plattner Institute at University of Potsdam,
August-Bebel-Strasse 88, Potsdam 14482, Germany
faeckert@gmail.com , marianalaraneves@gmail.com

Abstract

Question answering (QA) systems usually rely on advanced natural language processing components to precisely understand the questions and extract the answers. Semantic role labeling (SRL) is known to boost performance for QA, but its use for biomedical texts has not yet been fully studied. We analyzed the performance of three SRL tools (BioKIT, BIOSMILE and PathLSTM) on 1776 questions from the BioASQ challenge. We compared the systems regarding the coverage of the questions and snippets, as well as based on pre-defined criteria, such as easiness of installation, supported formats and usability. Finally, we integrated two of the tools in a simple QA system to further evaluate their performance over the official BioASQ test sets.

1 Introduction

Question answering (QA) is one of the most complex applications of natural language processing (NLP). QA systems need to precisely understand questions, in order to infer which information is being requested, and usually include steps such as question type and expected answer detection (Athenikos and Han, 2010; Neves and Leser, 2015). Likewise, the candidate documents or snippets that potentially contain the answers also need to be analyzed to extract the requested answer. Therefore, such systems usually rely on various NLP components, such as named-entity recognition, part-of-speech tagging and semantic parsing (Athenikos and Han, 2010).

Semantic role labeling (SRL) is one of the most popular tools to support QA systems (Shen and Lapata, 2007). It consists of automatically identifying predicates and their arguments, the so-called predicate-argument structures (PAS). For instance, for the question *"How many genes does the human hoxD cluster contain?"*, BioKIT (Dahlmeier and Ng, 2010), an SRL tool for biomedicine, correctly identified the following PAS: the predicate *contains* and two arguments (Arg0 - *the human hoxD cluster* and Arg1 - *How many genes*).

SRL is known for its potential to boost QA performance when extracting PAS from both the question and the text (e.g., snippets of sentences). Ideally, the same (or semantically related) PAS should be found in both of them in order to effectively support QA applications (Shen and Lapata, 2007). Hence, a good coverage is an important requirement for a tool to be suitable for QA. For our given example question, one of the answer snippets provided by the BioASQ challenge (Tsatsaronis et al., 2015) was *The human HOXD complex contains nine genes HOXD1, HOXD3, HOXD4, HOXD8, HOXD9, HOXD10, HOXD11, HOXD12 and HOXD13, which are clustered from [...].* The following PAS was detected by BioKIT: the predicate *contains* and the arguments Arg0 *the human HOXD complex* and Arg1 *nine genes*. In this example, there is a perfect match between the predicates from the question and the snippet. Further, the values for the argument Arg0 are similar and could be considered as a match too. The answer *nine genes* is indeed contained in Arg1, which also matches the argument type of the question word of the sentence. This example demonstrates how QA systems can benefit from PASs that were automatically detected by an SRL tool. However, language is more complex than reflected in this example. Thus, besides performing SRL, further challenges arise to integrate SRL and gain significant advantages in QA systems.

We are not aware of a comprehensive evaluation of available SRL tools on the BioASQ dataset, which is the most comprehensive dataset on biomedical QA (Tsatsaronis et al., 2015). We

* Current address: German Federal Institute for Risk Assessment, Diedersdorfer Weg 1, Berlin 12277, Germany

Proceedings of the 2018 EMNLP Workshop BioASQ: Large-scale Biomedical Semantic Indexing and Question Answering, pages 11–21

investigated three SRL tools, two of which were specifically developed for the biomedical domain, namely, BioKIT (Dahlmeier and Ng, 2010) and BIOSMILE (Tsai et al., 2006), and one which is based on deep learning, i.e., PathLSTM (Roth and Lapata, 2016). The latter has neither been trained nor tuned to biomedicine but has recently achieved promising results on SRL. Our contribution in this work is three-fold: (i) we provide a comprehensive overview on SRL for biomedicine and QA; (ii) we perform a comparison of selected tools regarding pre-defined criteria based on hands-on experiments; and (iii) we evaluated the selected SRL tools on the BioASQ datasets regarding their PAS coverage and performance in a QA system.

In the next section we provide an overview on previous work on SRL for biomedical QA, followed by the methodology we defined for the selection, comparison and evaluation of the SRL tools. In section 4 we present our results and discussion, followed by the conclusions of this work.

2 Overview of SRL for biomedical question answering

SRL has been well researched in recent decades and various tools have been created in the meantime. In addition to the tools, researchers have proposed standards for PAS annotations, such as the PropBank annotation format with its corresponding corpus (Kingsbury and Palmer, 2003). This was the standard followed by most SRL tools, as mentioned in (Palmer et al., 2010). However, they also presented two other popular formats for the English language, with corresponding corpora: FrameNet (Baker et al., 1998) and VerbNet (Kipper-Schuler, 2005).

Various features have been explored when building SRL tools based on machine learning algorithms. In 2004, Hacioglu et al. published an SRL approach which was based on chunking (Hacioglu et al., 2004). They trained a Support Vector Machine (SVM) to perform a semantic chunk segmentation step and role labeling. In their publication, they presented a complex set of features, e.g., words and part-of-speech tags and named entities, and their annotations followed the PropBank format. In the same year, Xue et Palmer experimentally explored the influence of certain newly proposed features on SRL results (Xue and Palmer, 2004). They could achieve significant improvements, especially by including syntactic frame features.

First efforts on neural-based SRL came a couple of years ago. In 2016, Roth et Lapata presented a novel SRL model that improved results of previous state of the art SRL tools for the open domain (Roth and Lapata, 2016). They utilized neural sequence modeling techniques and put special focus on improving the detection of nominal predicates. Their evaluation showed that the novel SRL model reached F_1-scores of 87.9% for in-domain data and 76.1% for out-of-domain data, thus improving the state of the art in both categories. The presented SRL tool is called PathLSTM and is publicly available with an up-to-date model.

Recently, Marcheggiani et al. published another neural model for dependency-based SRL (Marcheggiani et al., 2017). By applying a syntax-agnostic model, they could almost keep with the state of the art for in-domain data (F_1: 87.6%) and surpassed PathLSTM for out-of-domain data (F_1: 77.3%). Still last year, Do et al. discussed the role of implicit SRL and their approach to meet corresponding challenges (Do et al., 2017). Traditional SRL systems usually focused on explicit argument labels while implicit SRL aims at also finding the implicit ones. They used a recurrent neural semantic frame model for learning probability distributions over semantic argument sequences and could hereby improve the state of the art for detecting implicit semantic role labels.

2.1 Semantic Role Labeling on Biomedical Text Corpora

Since the last decade, numerous efforts have been made to apply SRL techniques to the biomedical domain. In 2004, Wattarujeekrit et al. published PASBio (Wattarujeekrit et al., 2004), a PropBank extension for the domain of molecular biology. The PASBio corpus contained PASs for a limited set of 30 predicate stems. The corpus was specifically designed to support the extraction of events in molecular biology.

A couple of years later, Chou et al. presented BioProp (Chou et al., 2006), a corpus with PASs for the biomedical domain. It is composed of approximately 500 articles from the GENiA corpus (Kim et al., 2003) which were annotated with PASs. The resulting corpus was used to train the biomedical SRL tool BIOSMILE. Initially, it supported finding PASs for the 30 predicates introduced by BioProp. Later, the tool was extended

and trained to support a total of 82 predicates, which are listed on the tool's website.[1] For the publication on the BIOSMILE tool (Tsai et al., 2006), the authors compared the performance of the latter to their initial SRL tool, which was only trained on PropBank data from the newswire domain. Being tested on BioProp data, the initial SRL tool could only reach an overall F_1-score of 64.2% while BIOSMILE reached 87.1%.

Later on, in 2009, Barnickel et al. presented their biomedical SRL system called SENNA which was based on a neural network (Barnickel et al., 2009). They managed to outperform tools like BIOSMILE regarding processing time but could only reach a comparably small F_1-score of 54% in the biomedical domain. In the following year, Dahlmeier et al. published an article on domain adaptation for SRL in the biomedical domain and introduced their respective SRL tool: BioKIT (Dahlmeier and Ng, 2010). It was developed as an alternative to the lack of training data in the biomedical domain and to the expensiveness to create training datasets. The authors discuss why, in their opinion, the BioProp corpus alone was not sufficient to create a good SRL tool for biomedicine. One of the reasons they mentioned was that BioProp was limited to 30 predicates and that many PASs were not covered in the corpus. When training BioKIT, they relied on the 1,982 PASs from BioProp and another 90,000 PASs from PropBank. They evaluated six supervised domain adaptation algorithms and concluded that the InstPrune algorithm performed best and reached an F_1-score of 85.38%.

More recently, in 2015, Zhang et al. showed, that clinical SRL can also significantly benefit from integrating domain adaptation techniques (Zhang et al., 2015). They relied on PropBank and NomBank from the newswire domain and BioProp as their source domain datasets. For the target domain, they used a manually annotated clinical corpus. They compared and evaluated three state-of-the-art domain adaptation algorithms: instance pruning, transfer self-training and feature augmentation. Finally, in 2016, Zhang et al. published another study where they investigated how their domain adaptation techniques for the clinical domain would apply on top of different syntactic parsers and features (Zhang et al., 2016). The best F_1-score they could reach on their clinical test data was 71.41%.

[1] http://bws.iis.sinica.edu.tw/BioC_BIOSMILE/

2.2 Semantic Role Labeling for Question Answering

In the past, different experiments and approaches for integrating SRL to QA systems have been elaborated. Some of them were partially related to the biomedical domain.

Shen et Lapata published an extensive study on the contribution of SRL to open-domain factoid QA (Shen and Lapata, 2007). Based on their experiments, they proposed a combination of syntactic and semantic annotations for the answer extraction part of QA. In general, they showed that QA can benefit from SRL but they also found much potential for preferable improvements. They pointed out that coverage is a key factor to achieve benefits from SRL for QA.

For their EPoCare QA system, Nio et al. created a role identification system for clinical QA using the PICO format (Niu et al., 2003). This role identification system showed similarities to the SRL task but was limited to a small set of task-specific roles and heavily based on the PICO format. Also in the biomedical domain, Shi et al. utilized SRL for their biomedical QA system for summary type questions (Shi et al., 2007). They basically used semantic role labels to measure semantic conformities in their sentence candidate ranking procedure. Therefore, they analyzed to which extend a sentence and the particular question contained matching PASs.

The biomolecular QA system by Lin et al. utilized BIOSMILE for detecting PAS both in the questions and in answer candidate sentences (Lin et al., 2008). The core of their QA system was a ranking module. Their results indicated that BIOSMILE in combination with named entity recognition is well suited for improving biomolecular QA systems. Nevertheless, biomolecular QA is a rather restricted domain with regard to biomedical QA. Finally, in the scope of the BioASQ challenge, our team experimented with the BioKIT tool for all four types of questions (factoid, list, yes/no and summary) (Neves et al., 2017). But the results we obtained with our simple approach were not very successful.

Type	No. questions	No. snippets
factoid	485	5,145
list	406	5,155
summary	392	4,069
yes/no	493	5,724
Total	1,776	20,093

Table 1: Statistics of the BioASQ training dataset for each question type.

3 Methods

In this section we describe the resources, tools and methodology that we used to select and analyze SRL tools for the biomedical QA.

3.1 BioASQ Dataset

We utilized the training dataset of 1,776 questions made available for the BioASQ challenge in 2017 (Tsatsaronis et al., 2015).[2] It combines test sets from the first four challenges. This dataset contains questions and the corresponding snippets of text which include the answer to the questions. The BioASQ dataset addresses four types of questions: factoid, list, summary and yes/no. Table 1 shows statistics on the number of questions and snippets. We considered all four question types in our analysis.

3.2 Criteria for the selection of SRL tools

Despite the many previous works on SRL for biomedicine (cf. Section 2), few tools are available for immediate use. Driven by time constraints, we decided to include the only two available SRL tools for biomedicine, i.e, BioKIT (Dahlmeier and Ng, 2010) and BioSMILE (Tsai et al., 2006), and one open domain SRL tool that has recently obtained state-of-the.art results, i.e., PathLSTM (Roth and Lapata, 2016). Due to the non-availability of an out-of-the-box working SRL tool based on their model, we did not include the tool developed by Marcheggiani et al. (Marcheggiani et al., 2017), even though it is freely available in GitHub. We give a short overview of the selected tools regarding their technical aspects:

BioKIT. It is available for the Linux operating systems and was mainly developed in Python and C.[3] BioKIT expects input as text files with line breaks as separators and outputs the SRL results in the CoNLL-09 format.[4] It can be built and compiled with Cmake if all required dependencies are previously installed.

BIOSMILE. It is available as a Web service and supports a REST API.[5] Requests via the API need to be in XML format and results are returned in the same format. As far as we know, the source code or binaries of BIOSMILE are not available.

PathLSTM. It is developed in Java, and the sources as well as a Java package are available in a GitHub repository.[6] It can be built via Apache Maven. Input and output formats are the standard ones for CoNNL.

3.3 Methodology for evaluation

We installed each tool (or accessed it via web service) and ran them on the questions and corresponding snippets of the BioASQ training dataset. The BIOSMILE API was rather slow and unstable, therefore, we did not manage to annotate the questions and snippets of the 4th year of the BioASQ challenge with BIOSMILE, which is part of the training dataset. Hence, we were only able to evaluate 1,308 questions and the corresponding 16,791 snippets for BIOSMILE. However, this should not significantly compromise the comparison between the tools, given that the BioASQ dataset appears to be very homogeneous.

We analyzed the tools with on three approaches: (a) an assessment based on pre-defined criteria (cf. Section 3.4); (b) an evaluation of the coverage by counting the numbers of questions and snippets for which PASs were found; and (c) performance of the tools in a simple QA system (cf. Section 3.5).

3.4 Evaluation criteria

We also analyzed the tools regarding some selected criteria:

Installation. It checks whether the tool could be easily installed or whether it required advanced skills for building, as well as whether we experienced any issues related to missing or outdated dependencies. This is important for a smooth integration into a QA system, given that the latter should not suffer from a lack of portability or maintainability after the integration.

[2] http://bioasq.org/

[3] http://nlp.comp.nus.edu.sg/software

[4] https://ufal.mff.cuni.cz/conll2009-st/task-description.html

[5] http://bws.iis.sinica.edu.tw/BioC_BIOSMILE/

[6] https://github.com/microth/PathLSTM

Support of standardized web API. It checks whether the tool offers a Web service and whether it could be accessed and used via API calls following standards, e.g., REST. This is important if no source or binaries are available to download. Additionally, this functionality constitutes a straightforward and easy way of integration without the use of own computational resources.

Input format. It specifies the supported standard input formats, e.g., XML, JSON or CoNLL. Standardized input formats can facilitate the integration process independent from the system's platform.

Output format. Similar to standardized input formats, it specifies the supported standard output formats, e.g., XML, JSON or CoNLL.

Parsing effort. It is our subjective rating on how easy it was to parse the content to and from the supported input and output formats.

Handling of special characters. It specifies whether the tool is able to handle special characters or if it runs into errors at presence of certain characters in the input text.

Speed. It assesses the tool's time performance for annotating questions and answer snippets. This should give an idea to which degree an integration of the respective SRL tool could slow down the whole QA system.

Robustness. It indicates how reliable the SRL tool behaves with regard to stability and accessibility. Issues with robustness might, for instance, be caused by the input or the unresponsiveness of a web service.

3.5 Integration of SRL tools into a QA system

We also evaluated the SRL tools in the context of a simple rule-based QA system. Our rules were designed to make use of SRL wherever possible, but we also included fall-back solutions for the case that no PAS were found (baseline system). We addressed three question types from the BioASQ challenge, namely yes/no, factoid and list questions. The rules and parameters were inspired and tuned by looking at the data from the first three years of the BioASQ challenge. Therefore, the evaluation of the SRL tools in our QA system was carried out only on the BioASQ dataset from the fourth year.

Table 2 gives an overview on different degrees of PAS matching in the rules for each question type. In general, the higher the level to which an answer snippet matches to a question, the higher is its relevance for the answer. More details on the rules that we defined for each question type are presented below.

Yes/No questions. In a first step, weights follow the matching schema in Table 2. In a second step, for each matching answer snippet with a relevance weight, we determined whether the answer is *yes* or *no* by analyzing the presence of negation terms close to the predicate. If a predicate was directly prefixed by the terms *"not"* or *"doesn't"*, its vote for the overall answer was *no* and received an initial weight boost of 1. If no negation terms were found in the answer snippet, the answer for this snippet was *yes*. Finally, the overall decision on the answer was decided by calculating the balance of the weighted *yes* and *no* votes. In case of no matching at all, the default answer is *yes* (fall-back solution).

Factoid questions. We focused on PASs whose predicate was present in the question and one argument that matched a question word, such as *"which"*,*"where"*, *"when"*, *"who"* or *"how"*. We followed the priority level from Table 2 by checking the matching predicates, argument types and contents. Candidate answers in the list were ordered according to the matching level. If there were no matching predicates between the answer snippets and the question, the list of answer candidates remained empty (no fall-back solution).

List questions. For list questions, and similar to factoid questions, we implemented a priority queue to detect arguments that probably contain the answer. The major difference between factoid and list questions is that list questions do not simply require a simple fact but an enumeration of facts that are relevant for the answer. This is taken into account by putting special attention on the recognition of symbols or words that usually indicate the presence of an enumeration inside the answer snippets. Therefore, we split the text of the arguments by commas and semicolons, as well as by the symbol '&' or the token *and*. Finally, if no predicate or PAS matches was found in any answer snippet, the system searched for any enumerations it could find (fall-back solution).

Priority level by PAS matching degree	Conditions per question type		
	yes/no	factoid	list
1	-	no matching predicate	
2	matching predicate		
3	level 2 + matching argument type		
4	level 3 + matching argument content		
Boosting Factors at each level	-	Argument type match with question word	
			Presence of enumerators

Table 2: Overview on the PAS matching levels and corresponding weights for the various question types.

Criteria	BioKIT	BIOSMILE	PathLSTM
Installation	very hard	-	hard
Web API	no	yes	no
Input format	text	XML	text
Output format	CoNLL	XML	CoNLL
Parsing effort	high	normal	high
Spec. charact.	bad	good	bad
Speed	fast	variable	very fast
Robustness	stable	unstable	stable

Table 3: Comparison of the three SRL tools regarding the selected criteria.

.

4 Results and Discussion

In this section we provide an assessment of the tools regarding the pre-defined criteria, the PAS coverage and the QA integration. For all approaches, we provide a comprehensive discussion based on our hands-on experiments with the tools.

4.1 Evaluation by criteria

We present an evaluation of the three selected SRL tools on the previously defined criteria (cf Table 3) and provide a detailed discussion on our impressions for each tool.

BioKIT. It does not support a binary, executable package nor a Web service and, therefore, it needed to be built on the Linux operating system. It was admittedly very hard to build and compile BioKIT, given that it is mainly written in Python and C but also depends on other packages and languages, such as Fortran. Many dependencies were outdated or missing and had to be searched in the Web. Therefore, simply following the installation instructions was not sufficient as some of the dependencies were themselves hard to build. Further, parsing the CoNLL format was more challenging than parsing XML or JSON into an object-oriented representation because the PASs had to be extracted by dynamically matching row and column indexes of the presented predicates and arguments. Additionally, BioKIT failed at handling special characters which led to annoying runtime errors. Usually, BioKIT's preprocessing pipeline was meant to eliminate problematic characters but some symbols (e.g., "æ", "ö" or "ȯ") were not handled by the system. As a result, BioKIT crashed with an error after processing thousands of sentences without returning any result when it ran into a special character. We collected a set of almost 20 of such characters that we eliminated in an own script-based preprocessing step. Depending on the length of the question or snippet, the processing of one question or snippet took at least 600 milliseconds or few seconds. This could be rated as a fast performance, but only when labeling many questions at once. If BioKIT was just used to process a single question, its runtime exceeded one minute, given the necessary time to load models into memory. In spite of the problem with special characters, we found BioKIT to be reliable and stable.

BIOSMILE. It is not available to download in any way (source code, binaries or executables). Therefore, we accessed it via a Web service with the REST API. The input and output were both formatted as XML, which facilitated parsing with standard XML parsing libraries. Further, we experienced no problems regarding special characters. However, with regard to the processing speed, the web service was rather unstable. In rare cases, it was possible to annotate a sentence in about a second but there were many problems regarding the robustness of the service. Frequently, it was not possible to send more than five requests in a row without waiting several minutes in between, otherwise the Web service became unresponsive for a long time. At some point, the service became so slow and had so many down times that we did not manage to annotate the data of the 4th year of the BioASQ challenge.

PathLSTM. Installing PathLSTM was not as hard as BioKIT but there were still some time-consuming issues. The tool can be build via Maven but it was under development during the

time that we were using it (as of June/2017). The code actually contained missing or wrong Maven dependencies and even a bug due to an outdated or not committed class. Hence, the installation process required certain research and code review to find out that an earlier git commit was working properly. Recently, the developers of PathLSTM published a more stable package but we did not check its feasibility yet. The input and output formats also followed the CoNLL format and, hence, were very similar to BioKIT. The CoNLL parser for BioKIT could be reused with small adaptations. PathLSTM had similar issues with special characters and could therefore reuse the preprocessing script that was created for BioKIT. When annotating many questions or answer snippets at once, PathLSTM could reach an annotation rate as low as 300 milliseconds per sentence. We considered it as being very fast, in comparison to the other tools. But if trying to annotate a single sentence, PathLSTM had the same issue as BioKIT and needed almost one and a half minute to load the models into memory.

4.2 Evaluation by predicate-argument structure coverage

This section compares the three SRL tools with regard to the usefulness and completeness of the detected PASs for the QA task. As pointed out in (Shen and Lapata, 2007), the PAS coverage of SRL systems is an important factor when trying to successfully integrate SRL into QA. Therefore, we compared the PAS coverage of each tool for questions and answer snippets from the BioASQ datasets. With special regard to the QA task, we analyzed the PAS matching coverage between questions and corresponding answer snippets. The PAS matching coverage is defined as the percentage of questions for which a PAS match could be found in any of the corresponding answer snippets.

PAS coverage for questions and answer snippets. Table 4 gives an overview on the PAS coverage reached by the respective tools for the various types of questions and for all answer snippets in general. Answer snippets are not presented by question types because they do not differ by question type. When comparing the coverage of different question types, the lowest coverage values were reached for summary questions, while the highest coverage values were reached for yes/no questions. Only BIOSMILE performed better on factoid and list questions than on yes/no questions. This is probably due to predicate stems such as *do* or *have* that are widely used in yes/no questions which not part of the predicates supported by BIOSMILE. BIOSMILE obtained the lowest coverage results of the three tools, especially when looking at the answer snippets, which only 15.2% of them had PAS annotations. For list questions, BIOSMILE reached a coverage of 65.1%, which was slightly higher than the coverage of BioKIT in the same category (61.8%). The main reason for the low coverage of BIOSMILE is most probably the limited set of 82 biomedical predicates.

BioKIT reached the maximum coverage for yes/no questions (99.8%). This could be due to the fact that *do* (727) and *be* (247) are the top predicate stems detected in the questions. In comparison to this, PathLSTM only labeled *do* 27 times as a predicate and never labeled *be*. Additionally, BioKIT also labeled auxiliary verbs as predicates, which appear very often in yes/no questions, and might explain its high coverage. Unfortunately, auxiliary verbs like *has* or *has been* are not known to have much semantic value. Hence, this high coverage might not be seen as an advantage for PathLSTM.

In general, PathLSTM obtained significantly higher coverage values than the other tools. In contrast to leaving out auxiliary verbs, the high coverage of PathLSTM can be explained by detecting about three times as many distinct predicates as BioKIT. In general, reaching a higher coverage might be good for QA, but by looking at some of the annotations, we found that PathLSTM labeled many nouns (as predicates) that did not even had in a verb form and most likely did not represent a predicate. For example, the most frequent predicates found by PathLSTM were nouns such as *disease* or *syndrome*, none of which are regularly used as predicates.

PAS matching coverage between questions and answer snippets We evaluated two levels of PAS matching coverage between questions and answer snippets. The first level, which is presented in Table 5, is the proportion of questions for which any answer snippet contained a PAS with the same predicate stem. The second level, which is presented in Table 6, requires that both predicate argument structures, from the question and from the particular answer snippet, share a similar argument type besides the predicate stem. The

Type	questions			answer snippets		
	BioKIT	BIOSMILE*	PathLSTM	BioKIT	BIOSMILE*	PathLSTM
factoid	61.4	49.5	96.3	88.5	15.2	98.5
list	61.8	65.1	96.1			
summary	39.5	19.3	90.3			
yes/no	99.8	42.7	96.1			

Table 4: PAS coverage (in %) for the various types of questions and for answer snippets. * BIOSMILE was only evaluated on data from the first three years of the BioASQ challenge.

Type	BioKIT	BIOSMILE*	PathLSTM
factoid	28.9	2.1	68.7
list	33.7	2.1	82.5
summary	16.3	0.7	70.9
yes/no	38.9	4.0	79.3

Table 5: Coverage of the questions (in %) for which a predicate match between the question and any of the related answer snippets was found. * BIOSMILE was only evaluated on data from the first three years of the BioASQ challenge.

Type	BioKIT	BIOSMILE*	PathLSTM
factoid	26.8	2.1	59.6
list	33.3	2.1	72.2
summary	13.8	0.7	60.5
yes/no	36.7	4.0	72.6

Table 6: Coverage of the questions (in %) for which a PAS match between the question and any of the related snippets was found. A PAS match was counted, if the predicate stem and any of the related argument types matched. * BIOSMILE was only evaluated on data from the first three years of the BioASQ challenge.

argument type and the predicate stem have to be related by the same predicate.

On both PAS matching levels, BIOSMILE reached very poor results, between 2% and 4% of PAS matching coverage. It appears that the same questions of Table 5 found by BIOSMILE also fulfill the requirements of Table 6, which might indicate that the found PAS matches are of a good quality. Nevertheless, the coverage is so low that BIOSMILE might only be considered in combination with other tools with a higher coverage in order to be efficiently used for biomedical QA. It would be pointless to exclusively rely on a tool that can only contribute to answering up to 4% of the questions.

In contrast, PathLSTM reached the highest PAS matching coverage values on both levels. Table 5 shows that PathLSTM obtained 82.5% PAS matching coverage for list questions with matching predicate stems. Further, Table 6 shows that for 72.2% of the questions, a matching argument type was present. On the one hand, the high coverage of PathLSTM might lead to a high recall when implementing a QA system on top of the annotations. On the other hand, our previous analysis of PathLSTM's predicates (cf. above) showed that they might be of poor quality.

The PAS matching coverage for BioKIT were not as high as the results reached by PathLSTM but superior than those from BIOSMILE. BioKIT leaves some space for improvement regarding coverage by finding matches for about one third of the factoid, list and yes/no questions and less than one sixth for summary questions. It is striking that the differences between the PAS matching coverage values of both levels are not very large (below 2.5%). In contrast to PathLSTM, this might be an indicator that PAS matches found by BioKIT are actually of a good quality and semantically relevant, and not just simply include matching terms that are not even real predicates and hence have no related arguments. Finally, PathLSTM reached PAS matching coverage values which are in average more than twice as large as those from BioKIT, but the quality and usefulness of the PAS matches from PathLSTM are still dubious.

4.3 Evaluation on the rule-based QA system

We compared BioKIT and PathLSTM regarding their performance on the fourth year of the BioASQ challenge. This dataset is composed of five batches of 100 questions and we provide detailed results for each batch. The results were obtained by uploading JSON result files to the BioASQ Oracle evaluation system[7]. The BIOSMILE system was not further evaluated due to (i) its low PAS matching coverage (cf. Table 5), which were very unpromising, and (ii) the instability of the Web service, which did not allow us to obtain results for this test set.

[7] http://participants-area.bioasq.org/accounts/login/?next=/oracle/

Batch	BioKIT	PathLSTM	NOSRL
1	96.43		
2	90.63		
3	96.0		
4	90.48		
5	100.0		
Average	94.71		

Table 7: Evaluation of the accuracy (in %) for yes/no questions on the 5 batches of the fourth BioASQ challenge.

Batch	BioKIT	PathLSTM
1	8.97	1.79
2	1.62	0
3	5.13	0
4	6.72	1.61
5	1.36	0.76
Average	4.76	0.83

Table 8: Evaluation of the mean reciprocal rank (in %) for factoid questions on the 5 batches of the fourth BioASQ challenge.

To measure the impact of the SRL components added to the QA system, we included a baseline QA solution (NOSRL) which did not rely on SRL but simply only on the fall-back solutions (cf. Section 3.5). As we could not propose appropriate fall-back solutions for the factoid questions, we evaluated the NOSRL baseline only for yes/no and list questions.

Yes/no. Table 7 evaluates the accuracy of the SRL-based QA and the NOSRL systems. The latter simply answered *yes* to all questions. For all 5 batches of the fourth year of the BioASQ challenge, the SRL-based systems did not provide any answers different than *yes*. Therefore, all systems achieved the same accuracy values. Obviously, our rules failed to match any of the valid *no*-answers in the fourth year's dataset. Subsequently, we cannot provide insight with respect to the performance of the SRL tools.

Factoid. Table 8 compares the performance on factoid questions for the different SRL-based QA systems by means of the mean reciprocal rank measure (MRR). The results show that the BioKIT-based QA system performed much better than the PathLSTM-based version. For the second and third batch, the PathLSTM-based system did not find any correct answer.

List. Table 9 shows the mean average F-measure results for the five batches. In general, BioKIT performed better than the NOSRL system, while the PathLSTM-based system performed worse than the latter.

Batch	BioKIT	PathLSTM	NOSRL
1	15.48	8.33	14.9
2	13.75	16.88	13.79
3	14.03	11.18	14.03
4	28.31	22.27	20.81
5	21.23	13.91	19.19
Average	18.56	14.51	16.54

Table 9: Evaluation of the mean average F-measure (in %) for list questions on the 5 batches of the fourth BioASQ challenge.

5 Conclusions and future work

Our experiments showed that BioKIT is the most suitable SRL tool for biomedical QA, and in a lesser degree, PathLSTM might also be considered. For both tools, different challenges might arise for their integration. While BioKIT still has a lack of coverage, PathLSTM probably detects too many PAS candidates and therefore performed poorly in our simple QA system. A first approach to increase the precision for PathLSTM would include filtering out noun predicates which do not have a verb stem. We would like to perform a more comprehensive evaluation of the quality of the PAS, given that we only carried out a small validation of a few of them. Recently, a new SRL tool (He et al., 2018) has been published and should also be considered in future experiments.

While BioSMILE is readily available, its web service is unstable and the coverage is extremely low. BioKIT is hard to install, but provides a good coverage of PAS which is suitable for the QA task. We assume that PathLSTM is too generic, as it is an open-domain SRL tool. It might therefore have trouble to compete with specialized biomedical SRL on data from the biomedical domain. Finally, even though the coverage from PathLSTM is high, an analysis of some of its PAS shows that many predicates have no semantic meaning and many correspond to nouns which do not behave as predicates in the corresponding sentences.

References

Sofia J. Athenikos and Hyoil Han. 2010. Biomedical question answering: A survey. *Computer Methods and Programs in Biomedicine*, 99(1):1 – 24.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 36th Annual Meeting of the Associ-*

ation for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1, pages 86–90. Association for Computational Linguistics.

Thorsten Barnickel, Jason Weston, Ronan Collobert, Hans-Werner Mewes, and Volker Stümpflen. 2009. Large scale application of neural network based semantic role labeling for automated relation extraction from biomedical texts. *PLoS One*, 4(7):e6393.

Wen-Chi Chou, Richard Tzong-Han Tsai, Ying-Shan Su, Wei Ku, Ting-Yi Sung, and Wen-Lian Hsu. 2006. A semi-automatic method for annotating a biomedical proposition bank. In *Proceedings of the workshop on frontiers in linguistically annotated corpora 2006*, pages 5–12. Association for Computational Linguistics.

Daniel Dahlmeier and Hwee Tou Ng. 2010. Domain adaptation for semantic role labeling in the biomedical domain. *Bioinformatics*, 26(8):1098.

Quynh Ngoc Thi Do, Steven Bethard, and Marie-Francine Moens. 2017. Improving implicit semantic role labeling by predicting semantic frame arguments. *arXiv preprint arXiv:1704.02709*.

Kadri Hacioglu, Sameer Pradhan, Wayne H Ward, James H Martin, Daniel Jurafsky, et al. 2004. Semantic role labeling by tagging syntactic chunks. In *CoNLL*, pages 110–113.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369. Association for Computational Linguistics.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Junichi Tsujii. 2003. GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl 1):i180–i182.

Paul Kingsbury and Martha Palmer. 2003. Propbank: the next level of treebank. In *Proceedings of Treebanks and lexical Theories*, volume 3. Citeseer.

Karin Kipper-Schuler. 2005. *VerbNet: a broad-coverage, comprehensive verb lexicon*. Ph.D. thesis, Computer and Information Science Department, Universiy of Pennsylvania, Philadelphia, PA.

Ryan T. K. Lin, Justin Liang-Te Chiu, Hong-Jie Dai, Min-Yuh Day, Richard Tzong-Han Tsai, and Wen-Lian Hsu. 2008. Biological question answering with syntactic and semantic feature matching and an improved mean reciprocal ranking measurement. In *IRI*, pages 184–189. IEEE Systems, Man, and Cybernetics Society.

Diego Marcheggiani, Anton Frolov, and Ivan Titov. 2017. A simple and accurate syntax-agnostic neural model for dependency-based semantic role labeling. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 411–420, Vancouver, Canada. Association for Computational Linguistics.

Mariana Neves, Fabian Eckert, Hendrik Folkerts, and Matthias Uflacker. 2017. Assessing the performance of olelo, a real-time biomedical question answering application. In *BioNLP 2017*, pages 342–350, Vancouver, Canada,. Association for Computational Linguistics.

Mariana Neves and Ulf Leser. 2015. Question answering for biology. *Methods*, 74:36 – 46. Text mining of biomedical literature.

Yun Niu, Graeme Hirst, Gregory McArthur, and Patricia Rodriguez-Gianolli. 2003. Answering clinical questions with role identification. In *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine - Volume 13*, BioMed '03, pages 73–80, Stroudsburg, PA, USA. Association for Computational Linguistics.

Martha Palmer, Daniel Gildea, and Nianwen Xue. 2010. Semantic role labeling. *Synthesis Lectures on Human Language Technologies*, 3(1):1–103.

Michael Roth and Mirella Lapata. 2016. Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1192–1202, Berlin, Germany. Association for Computational Linguistics.

Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic*, pages 12–21. ACL.

Zhongmin Shi, Gabor Melli, Yang Wang, Yudong Liu, Baohua Gu, Mehdi M Kashani, Anoop Sarkar, and Fred Popowich. 2007. Question answering summarization of multiple biomedical documents. In *Advances in Artificial Intelligence*, pages 284–295. Springer.

Richard Tzong-Han Tsai, Wen-Chi Chou, Yu-Chun Lin, Cheng-Lung Sung, Wei Ku, Ying-Shan Su, Ting-Yi Sung, and Wen-Lian Hsu. 2006. BIOSMILE: adapting semantic role labeling for biomedical verbs: an exponential model coupled with automatically generated template features. In *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, pages 57–64. Association for Computational Linguistics.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, et al. 2015. An overview of the BIOASQ

large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16(1):138.

Tuangthong Wattarujeekrit, Parantu K Shah, and Nigel Collier. 2004. PASBio: predicate-argument structures for event extraction in molecular biology. *BMC bioinformatics*, 5(1):155.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *EMNLP*, pages 88–94.

Yaoyun Zhang, Min Jiang, Jingqi Wang, and Hua Xu. 2016. Semantic role labeling of clinical text: Comparing syntactic parsers and features. In *AMIA Annual Symposium Proceedings*, volume 2016, page 1283. American Medical Informatics Association.

Yaoyun Zhang, Buzhou Tang, Min Jiang, Jingqi Wang, and Hua Xu. 2015. Domain adaptation for semantic role labeling of clinical text. *J Am Med Inform Assoc*, 22(5):967–979. 26063745[pmid].