






Segmentation of Words Written in the Latin Alphabet: A Systematic Review

Marcelo A. Inuzuka^(✉), Acquila S. Rocha^(✉),
and Hugo A. D. Nascimento^(✉)

Instituto de Informática – Universidade Federal de Goiás (UFG),
Caixa Postal 131, Goiânia, GO 74.001-970, Brazil
{marceloakira,acquilarocha,hadn}@inf.ufg.br

Abstract. In this systematic literature review (SLR) we summarize studies that address the word segmentation problem (WSP) for Latin-based languages. We adopted the protocol of Kitchenham et al. for the review. The search in academic repositories found 771 works, from which 89 were selected. After a quality assessment step, 69 papers were chosen for data extraction. The results point to a divergence in terminology of this problem, two of which are more relevant, having specific techniques, corpus and application context: compound splitting and identifier splitting. We analyze the state of the art of each context, pointing out differences and similarities in approaches. We hope that these results can serve as a guide for future investigations and advancement of WSP.

Keywords: Natural language processing · Word segmentation · Identifier splitting · Compound splitting · Systematic literature review

1 Introduction

Word segmentation (WS) is a task of the natural language processing (NLP) area that consists of dividing a string into constituent parts for serving a given purpose. This task is similar to word tokenization, but differs as we will see more below. Depending on the linguistic context or the application domain, this task varies in taxonomy. In the present article, we perform a systematic literature review (SLR) of WS applied on texts written in the Latin alphabet.

The motivation of this work originated in experiences of the processing of legal texts in Portuguese. Due to errors in converting PDF file format to plain text, long spurious strings have emerged such as ‘decisãoanteriorjáservecomomandadodeprisãopreventivaeofício’ that should be corrected to ‘decisão anterior já serve como mandado de prisão preventiva’ (previous decision already serves as a warrant for custody). Looking for solutions to the problem, we found the *nltk.tokenize* library, which in turn has a sub-module *nltk.tokenize.stanford_segmenter*¹, but only supports Chinese and Arabic languages. In a prior exploratory research, we found some word segmentation tools

¹ Available at https://www.nltk.org/_modules/nltk/tokenize/stanford_segmenter.html.

in English with technical analysis, but without scientific benchmarks². These initial experiences motivated us to conduct a systematic literature review.

Word segmentation (WS) and word tokenization (WT) can be confused each other, as both produce sub-strings as a result. The difference is at the input strings and whether or not word delimiters (WDs) are supported, such as spaces or punctuation. In languages such as Portuguese or English, it is normal for the WT input string to be made up of several words separated by WDs and if not, WS can be used to get the tokens separated. In languages like Chinese, there are no WDs, so WS is most commonly employed. This way, WS can be used as a WT subtask if there is any string that needs to be segmented. Following, we focus on a formal description of the *word segmentation problem* (WSP) what can be defined as an optimization problem. A general formulation for it can be: given a string s , consisting of non-delimiting characters of words, find a split of s into a list of words $W = \langle w_1, w_2, \dots, w_n \rangle$, with $w_1 \cdot w_2 \dots \cdot w_n = s$ and $|w_i| \geq 1$, so that an objective function $f(W)$ is optimized and a set of constraints are satisfied. There is a considerable amount of different WSP definitions in the literature, each one with a particular aim and set of constraints. A common and simple specialization of the general formulation is to ignore f (or make it constant) and to ensure that every w_i belongs to a given dictionary. Another specialization of the problem is to find a segmentation W with minimum number of splits. This can be formalized as: *Minimize* $F(W) = |W|$ constrained to have every w_i belonging to a dictionary. It is also possible to deal with imprecision or errors in s . In that case, $f(W)$ could measure how the terms in W deviate from their most similar words in a dictionary. A usual constraint for that case would be to enforce that every word in W is at most k characters different from its closest valid word in the dictionary. Different WSP formulations, in general, demand distinct algorithmic approaches for proving a good solution.

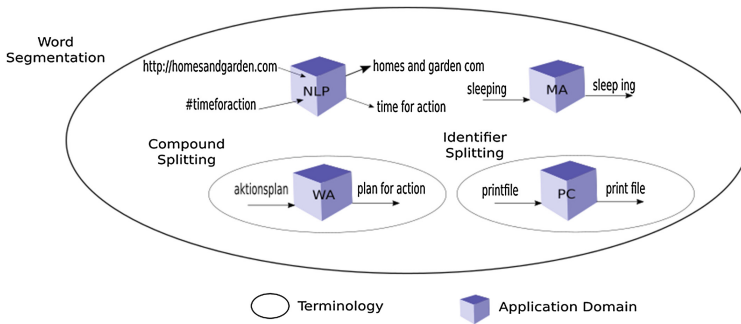


Fig. 1. Word Segmentation application domains

Word segmentation tasks also vary in method and in taxonomy according to the application domain, as seen in Fig. 1. **Word Alignment (WA)** is a task

² Example of tool: <http://www.grantjenks.com/docs/wordsegment/>.

in machine translation used for translating texts from a language to another. Languages that have a high amount of compounds, like German, make this task more difficult, because compounds has to be splitted to find corresponding words to the target language. For example: translate the German compound ‘aktionsplan’ to the English words ‘plan for action’. In these contexts, the WS task is called *compound splitting*. **Program comprehension (PC)**. In software engineering, WS is used to analyze source code by dividing identifiers such as variable names that can usually be divided into acronyms or understandable parts. For example: ‘printfile’ to ‘print file’. In this context, WS is called *identifier splitting*. **Social analytics (SA)**. In order to gain a better understanding of the Web, WS can be used to analyse hashtags and domain names (URLs). For example: ‘homesandgardens’ to ‘homes and gardens’. In this context, WS is also employed and can called *hashtag splitting* or *domain name splitting*, respectively. **Morphological analysis (MA)**. A word can be analyzed in morphemes in order to understand its formation. For example: sleep-ing, dis-member-ed, etc. WS is also used in this context [3]. **Natural language processing (NLP)**. This is the most general case, in which the input text has been affected by noise [2] such as typos, OCR errors, char-code conversion errors, speech-to-text conversion error, etc.

The methodological framework applied for the development of this work follows the recommendations of Kitchenham [4], which establish a sequence of steps for producing consistent, auditable, and reproducible systematic reviews. The methodology suggested by the authors involves three stages: creating a review protocol, conducting the review, and presenting the results. The following sections reflect this methodology.

2 Review Protocol

We now present the planning stage of the SLR methodology. This section is divided into 4 subsections. Section 2.1 establishes the review questions; Sect. 2.2 presents the keywords and the search strategies; Sect. 2.3 defines the inclusion and the exclusion criteria; and Sect. 2.4 defines a quality evaluation.

2.1 Research Questions

The main objective of the SLR was to answer the following question: “What is the state of the art in WS methods?”. Some more specific questions that unfolded the previous one were formulated: (RQ.1): What are the differences in WS methods in specific contexts? (RQ.2) Which technique performed best in specific contexts? (RQ.3) What is the state of the art in WS in the Portuguese language context?

2.2 Search Strategy

Conducting searches takes into account three primary factors: study sources, search keys, and scope delimitation. Nine study sources³ were chosen considering previous SLRs and informal conversations with literature review experts: ACM Digital Library (AC), arXiv (AX), Google Search (GO), Google Scholar (GS), IEEE Xplore (IX), Scopus (SC), SpringerLink (SL), Science Direct (SD) and Web of Science (WS). In order to formulate search criteria, we separated the search into three types of aspects: search elements (SE - Table 1), search restrictions (SR - Table 2) and search filters (SF - Table 3). A search string (SS) consists of a combination of SE, and finally of using SR and SF to limit the results, as showed in Table 4.

Table 1. Search elements

Reference	SE - Search elements
SE1	<i>'compound splitting' OR 'identifier splitting'</i>
SE2	<i>'word segmentation'</i>
SE3	<i>'natural language processing' OR 'NLP'</i>
SE4	<i>'segmentação' OR 'separação' OR 'segmentação lexical' OR 'processamento de palavras compostas' OR 'análise léxica'</i>
SE5	<i>'segmentação de palavras'</i>
SE6	<i>'processamento de linguagem natural' OR 'PLN'</i>
SE7	<i>'palavras compostas' OR 'palavras coladas' OR 'palavras grudadas'</i>

Table 2. Search restrictions

R1=philosophical, R2=education, R3=chemistry, R4=gear, R5=mechanical, R6=biomedical, R7=engineering, R8=optical, R9=pharmacologic, R10=pharmaceutic, R11=surgery, R12=organic, R13=alloy, R14=biochemical, R15=physics, R16=molecular, R17=disorders, R18=medical, R19=urology, R20=energy, R21=cardiology, R22=clinical, R23=simulation, R24=radio, R25=chemical, R26=philosophy, R27=cultural, R28=psychology, R29=chinese, R30=urdu, R31=thai, R32=vietnamese, R33=myanmar, R34=khmer, R35=arabic, R36=jobs, R37=tibetan, R38=ad, R39='call for papers', R40=japanese, R41=ocr, R42=biologic, R43=handwritten, R44=burmese, R45=infant, R46=lao, R47=geoscience, R48=javanese, R49='question answering'

It is necessary to apply search restrictions in order to limit the amount of search results to a viable number of works to read. For example, in the Table 2, we

³ Study sources details is documented at JSON file: <https://git.io/Je0DZ>.

use search restrictions to eliminate results outside the desired domain (education, philosophical, etc.) and language (chinese, thai, etc.).

Table 3. SF - Search filters

Reference	Elements
F1	Published from 2014 to 2019
F2	Search content in English
F3	Search content in Portuguese
F4	Science computation area
F5	From first 200 best ranked results
F6	Ad-hoc assessment

For each search engine, one or two searches were performed. This was necessary due to the large amount of results in some specific searches and limitations of the search string length. To facilitate the documentation of the searches, a database in JSON format has been edited⁴, as well as a bash script has been created⁵. These components allow to generate the desired search strings. For example, for repeating the search IX2 - second search in the IEEE Xplore Digital Library, we can execute the command ‘gen-search-string’ as described in Fig. 2.

```
$ ./gen-search-string.sh SE2 SE3 R18 R29 R30 R31 R32 R34 R35 R37 R43
( "word segmentation") AND ( "natural language processing" OR "NLP") -medical
-chinese -urdu -thai -vietnamese -khmer -arabic -tibetan -handwritten
```

Fig. 2. Generating a search string with a bash script

Note that only SE and SR items were combined, since the SF value for the example above is ‘None’. In searches that have filters it is necessary to apply them in the web interface of the digital library. For example, the GS2 search has filters F1 and F2. So, it is necessary to select the options ‘publish from 2014 and 2019’ and ‘search content in English’ (see the Table 3). With this approach, it was possible to experiment and apply different search strings in an efficient way.

2.3 Inclusion/Exclusion Criteria

Inclusion and exclusion criteria were defined for guiding the selection of relevant studies. For a study to be selected, we considered that all inclusion criteria should be met, as well as not meeting any exclusion criteria.

⁴ Details of search strings are also available at: <https://git.io/Je0DZ>.

⁵ A bash script file was created to generate search strings: <https://git.io/Je0DC>.

Table 4. SS - Search strings

Search	Elements	Restrictions	Filters	N. of results
GS1	$SE1 \wedge SE3$	{R2, R3, R5, R6, R7, R29}	F1	114
GS2	$SE2 \wedge SE3$	{R29, R30, R31, R32, R33, R34, R35, R36, R37, R18, R40, R43, R45, R41}	$F1 \wedge F2$	57
GS3	$SE4 \wedge SE6 \wedge SE7$	None	F3	135
GS4	$SE5 \wedge SE6$	None	F3	30
SD1	SE1	{R7, R8, R9, R10, R11, R3, R12, R13, R14, R15, R16, R17, R18, R19, R20, R21}	None	24
SD2	$SE2 \wedge SE3$	{R29, R30, R31, R32, R33, R34, R35, R37, R18, R40, R28, R41, R42}	None	11
IX1	SE1	None	None	11
IX2	$SE2 \wedge SE3$	{R29, R30, R32, R31, R34, R35, R37, R18, R43}	None	72
AC1	SE1	None	None	43
AC2	$SE2 \wedge SE3$	{R29, R30, R31, R32, R34, R35, R18, R43, R44, R33, R45, R40}	None	26
SC1	SE1	None	F4	59
SC2	$SE2 \wedge SE3$	{R29, R30, R31, R32, R46, R35, R18, R43, R33, R37, R44, R40, R47, R48, R49}	F4	69
WS1	SE1	None	None	30
WS2	$SE2 \wedge SE3$	{R29, R18, R43, R40, R32, R35, R37, R30, R31, R33}	None	26
SL1	SE1	{R22, R23, R24, R25, R7, R3, R26, R18, R27, R15, R28}	None	39
SL2	$SE2 \wedge SE3$	{R29, R18, R43, R40, R32, R46, R35, R37, R30, R31, R33, R41, R7, R45, R2, R49}	None	17
AX1	SE2	{R29, R35, R40, R45, R32, R34, R37, R43}	F6	56
GO1	$SE2 \wedge SE3$	{R29, R30, R31, R32, R33, R34, R35, R36, R37, R38, R18, R39}	$F5 \wedge F6$	10

In this sense, we chose the following inclusion criteria: (IC1) having full text available; (IC2) having an abstract; (IC3) being written in English or in Portuguese; (IC4) being a scientific study or a grey literature. As scientific studies, we considered papers, technical reports, surveys, master dissertations and doctoral thesis. As grey literature [4], we included technical reports, preprints, work

in progress, software repositories with source codes, and documentations in web portals. For the later, we accepted web portals with relevant publication volume and with good evaluation from their users, or simply by an *ad-hoc* assessment. The exclusion criteria were: (EC1) not addressing WS; (EC2) addressing specific African language studies; (EC3) addressing specific Asian language studies.

2.4 Quality Assessment (QA)

The following quality assessment questions were devised: (QA1) Are the research context described in the study? (QA2) Is the research methodology clearly explained in the study? (QA3) Is data and performance analysis evidently explained in the study?

For each question, three possible answers were established - Yes, Partially, and No. These answers were assigned to a score of 1, 0.5 and 0.0, respectively. Thus, each study could reach a maximum of 3.0 points and a minimum of 0.0 points. All studies below 2.0 points were disqualified (excluded).

3 Conducting the Review

By using the search strings, we found and downloaded the resulting references in the BibTeX format⁶. All digital libraries exported to this format except Springer-Link (SL), which references were available only in CSV and had to be converted to BibTeX using the *csv2bib* tool⁷. The SLR was managed using Parsifal⁸ that, in addition to importing the BibTeX items, also supported reference duplicate detection, selection, classification and data extraction.

In the selection stage, 771 studies were obtained as candidates. By reading the title and the abstract (when available) of each study, 604 papers were rejected, 89 were detected as duplicates, and 78 were approved. In the classification stage, from the 78 selected papers, 9 were eliminated with a score equal to or below 2.0 points and 69 were classified for data extraction. The data extraction step used a form created according to the Table 5. At this stage, it was necessary to download the full text of all classified studies for a complete reading. The Zotero software⁹ was employed for managing and sharing these texts.

The data extracted from the studies at the last step is shown in Table 6, with references available in a BibTeX file¹⁰.

4 SLR Results

In this section, we answer the research questions, based on the extracted data.

⁶ Bibtex is used in LaTeX documents to describe references: <http://www.bibtex.org/>.

⁷ *csv2bib* converts CSV to bibtex. See <https://github.com/jacksonpradolima/csv2bib>.

⁸ Parsifal is a web software for managing SLR. <https://parsif.al/>.

⁹ Zotero helps to collect and organize research references. <https://www.zotero.org/>.

¹⁰ The complete references in Table 6 can be downloaded at <https://git.io/Je0D8>.

Table 5. Data extraction form

Description	Type	Values
Q1. What type of corpus is used?	Select one	G=Generic, PC=Parallel Corpus, SD=Specific Domain, O=Other
Q2. What is utilization domain?	Select one	GU=General Use, MT= Machine Translation, OU=Other Use, SE=Software Engineering, SN=Social Network Analysis
Q3. What languages did the study address?	Select many	C=Chinese ^a , E=English, F=French, G=German, O=Other, P=Portuguese, S=Spanish, U=Universal, V=Various
Q4. What terminology is used?	Select one	CS=Compound Splitting, IS=Identifier Splitting, WS=Word Segmentation
Q5. What type of corpus is tested?	Select one	G=General, H=Hashtag, I=Identifiers, C=Compounds, U=URL
Q6. Did the work involve deep learning technique? Which one?	Select many	LSTM=Bi/Long Short-Term Memory, RNN=Bi/Recurrent Neural Network, CNN=Convolutional Neural Network, GRU=Gated Recurrent Units, N=No, O=Other, SNLM=Segmental Neural Network Model, T2T=Tensor2Tensor MTM
Q7. Did the work use word embedding? Which ones?	Select one	CE=Char Embedding, N=No, O=Other, WE=Word Embeddings
Q8. Did the work use different techniques from deep learning? Which one?	Select many	CRF=Conditional Random Field, DP=Dynamic Programming, WD=Word Dictionary, EA=Expand Abbreviations, LA=Lexical Analysis, MA=Morphological Analysis, MS=Morpheme Segmentation, NA/NI=Not Applicable/Not Informed, O=Other, N=NGRAM, PT=Pos Tagging, ST=Statistic Techniques, SW=Stop Words List, TE=Text Entailment, VA=Viterbi Algorithm

^a Chinese is not the primary language evolved in this works

4.1 RQ.1: What Are the Differences in WS Methods in Specific Contexts?

According to the data survey, when considering the use of the term ‘compound splitting’ as a specific context of the WS task for segmenting compound words, we obtained 21 studies: 7, 9, 13, 14, 16, 18, 19, 21, 22, 26, 27, 28, 30, 35, 37, 47, 49, 51, 52, 53, 56, 58, 69 – see Table 6. This represents 34.7% of the total amount of papers. There is no occurrence of usage of deep learning techniques in these studies. The most used methods are based on statistical techniques (ST), morphological analysis (MA) and lexical analysis, appearing in 7, 5 and 4 studies, respectively. In the context of this problem, the German language (G) was the one with the highest number of occurrences, as well as in the machine translation application (MT).

In the context of identifier splitting, 14 studies were found (20% from the total): 2, 4, 5, 8, 11, 24, 29, 38, 39, 50, 54, 57 and 59. The techniques of word dictionary (WD) and expand abbreviations (EA) appeared in 4 and 2 studies respectively. Deep learning (DL) was used in two works (29 and 54), and the most used language was English, in all occurrences.

Table 6. Data Extracted from the studies

N	Reference	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
1	(Smith et al. 2018)	G	G	U, V	WS	G	LSTM, RNN	CE	CRF
2	(Binkley et al. 2013)	SD	ES	E	IS	I	N	N	NA/NI
3	(Aken et al. 2011)	O	G	E	WS	G	RNN	N	ST
4	(Hill et al. 2014)	SD	ES	E	IS	I	N	N	NA/NI
5	(Guerrouj et al. 2014)	SD	ES	E	IS	I	N	N	NA/NI
6	(Wang et al. 2015)	G	G	E	WS	G	N	N	ST
7	(Shishkova et al. 2016)	SD	T	G, E	CS	C	N	N	O
8	(Guerrouj et al. 2010)	SD	ES	E	IS	I	N	N	
9	(Lee et al. 2007)	G	G	E, V	CS	G	N	N	ST, NG
10	(Wang et al. 2015)	G	G	E	WS	U	N	N	ST
11	(Dit et al. 2011)	SD	ES	E	IS	I	N	N	O
12	(Doval et al. 2018)	G	G	V	WS	G	RNN	N	NG
13	(Kraaij et al. 1998)	O	G	O	CS	C	N	N	NA/NI
14	(Ordelman et al. 2003)	SD	G	E, O	CS	C	N	N	NA/NI
15	(Shao et al. 2017)	SD	G	V	WS	G	RNN, GRU	N	VA, NG, MS
16	(Khaitan et al. 2009)	G	O	E	CS	U	N	N	ST, NG, SW
17	(Liang et al. 2014)	SD	G	V	WS	G	N	N	NA/NI
18	(Henrich et al. 2011)	SD	O	G, E	CS	C	N	N	MA
19	(Rigouts Terry et al. 2016)	SD	T	O	CS	C	N	N	O, PT
20	(Baziotis et al. 2019)	SD	G	E	WS	H	RNN	N	NA/NI
21	(Koehn et al. 2003)	PC	T	G, E	CS	C	N	N	ST, LA
22	(Jagfeld et al. 2017)	PC	G	G	CS	C	N	N	TE, LA
23	(Garbe et al. 2019)	O	G	E	WS	G	N	N	DP, O
24	(Carvalho et al. 2015)	SD	ES	E	IS	I	N	N	NA/NI
25	(Hewlett et al. 2011)	G	G	U	WS	G	N	N	NA/NI
26	(Sugisaki et al. 2018)	SD	G	G	CS	C	N	N	MA, LA
27	(Escartín et al. 2014)	PC	T	G, S	CS	G	N	N	PT
28	(Alfonseca et al. 2008)	PC	T	G, E	CS	C	N	N	ST
29	(Li et al. 2018)	SD	ES	E	IS	I	CNN, LSTM	N	CRF
30	(Fritzingert et al. 2010)	O	T	G	CS	C	N	N	MA, ST
31	(Johnson et al. 2009)	G	G	O	WS	G	N	N	ST
32	(Chen et al. 2016)	G	G	V	WS	G	N	N	NA/NI
33	(Paul et al. 2011)	PC	G	C, E, V	WS	G	N	N	O
34	(Paul et al. 2009)	PC	T	C, E, V	WS	G	N	N	O
35	(Macherey et al. 2011)	PC	T	G, E, V	CS	C	N	N	MA, DP, O
36	(Kawakami et al. 2018)	G	G	C, E	WS	G	LSTM, SNLM	N	LA, O
37	(Ma et al. 2016)	SD	T	G, U	CS	C	N	N	NA/NI
38	(Corazza et al. 2012)	SD	ES	E	IS	I	N	N	DW, EA, O
39	(Enslin et al. 2009)	SD	ES	E	IS	I	N	N	EA, DW
40	(Macháček et al. 2018)	G	T	G, V	WS	G	T2T	O	MA
41	(Moreau et al. 2019)	G	G	U	WS	G	RNN	N	NA/NI
42	(Sennrich et al. 2015)	G	T	V	WS	G	N	N	NA/NI
43	(Yang et al. 2017)	G	G	C	WS	G	LSTM	N	NA/NI
44	(Jenks et al. 2019)	G	G	E	WS	G	N	N	
45	(Reuter et al. 2016)	G	R	E, P	WS	H	N	N	PT, NG, O
46	(Srinivasan et al. 2012)	O	R	E	WS	U	N	N	ST
47	(Cöster et al. 2004)	SD	O	O	CS	C	N	N	O
48	(Wu et al. 2012)	SD	G	E	WS	G	N	N	O

(continued)

Table 6. (*continued*)

N	Reference	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
49	(Weller-Di et al. 2017)	SD	T	G	CS	C	N	N	NA/NI
50	(Hucka et al. 2018)	G	ES	E	IS	I	N	N	NA/NI
51	(Daiber et al. 2015)	O	T	O	CS	C	N	WE	NA/NI
52	(Shapiro et al. 2016)	G	G	O	CS	G	N	N	NG
53	(Clouet et al. 2014)	G	G	E, V	CS	C	N	N	ST
54	(Markovtsev et al. 2018)	SD	G	V	IS	I	LSTM, CNN	N	O
55	(Popović et al. 2006)	G	G	V	WS	G	N	N	PD, NG
56	(Norvig et al. 2019)	SD	T	G, E	CS	G	N	N	ST
57	(Guerrouj et al. 2013)	SD	ES	E	IS	I	N	N	DW, O
58	(Ziering et al. 2016)	G	O	G, V	CS	C	N	N	MA, PT, O
59	(Guerrouj et al. 2012)	SD	ES	E	IS	I	N	N	DW, O
60	(Shao et al. 2018)	G	G	C, F, E, P, V	WS	G	RNN	CE	NA/NI
61	(Kazakov et al. 2001)	O	G	F	WS	G	NA	NA	MA, O
62	(Roshani et al. 2014)	G	G	U	WS	G	N	N	VA
63	(Tambouratzis et al. 2009)	O	G	V	WS	G	N	N	MA
64	(Sun et al. 2013)	G	G	U	WS	G	N	N	O
65	(Gabay et al. 2008)	G	G	O	WS	G	N	N	PT
66	(Wang et al. 2011)	G	G	E	WS	U	N	N	ST
67	(Hewlett et al. 2011)	G	G	U	WS	G	N	N	O
68	(Stahlberg et al. 2012)	PC	G	S, E, V	WS	G	N	N	O
69	(Owczarzak et al. 2014)	G	G	V	CS	G	N	N	LA

In the more general context, which uses the term ‘word segmentation’, the largest number of studies were found, 34 (49.3%) in total. In this context, DL techniques were more frequent, about 11 studies (32% of the total). When DL is employed, RNN and LSTM techniques prevail, with 7 and 3 occurrences. Otherwise, statistics, POS tagging (PT) and N-Grams (N) techniques are the most frequent ones, with 12, 5 and 5 occurrences respectively.

Figure 3(a) shows the number of the selected scientific production from 1998 to 2019 in each specific word segmentation context (WS, CS, IS). On average, since 1998, there was an increase of the number of studies in the three segmentation contexts. CS and WS received more publications at the period 2016–2019.

4.2 RQ.2: Which Technique Performed Best in Specific Contexts?

To obtain the state of the art of the WS techniques reliably, it is necessary to apply benchmarking on standardized corpus. Common corpus were found for the IS context, but there was no standardization when considering CS and WS.

In Fig. 3(b) we analyzed the occurrence of DL techniques from 1998 to 2019. We note that, since 2010, it has been an increase in DL and a decrease in the use of other approaches, denoting a certain interest of the scientific community in that technique. Thus, we can say that the use of DL is a trend in recent years.

In the IS context, study 29 (see Table 6) presents a state of the art new technique based on deep learning, called CNN-BiLSTM-CRF, that outperformed other techniques such as LINSÉN, LIDS and DTW.

In the context of CS, there is no standardized corpus either. In general, metrics are based on the performance of CS usage applied in machine translation, where BLEU was the most used. However, Escartín [1] suggested a way to mediate CS performance using precision, recall and F-measure metrics.

In the context of WS, there is no standardized corpus. However, in studies 12 and 41, there is an attempt to establish comparative metrics, with precision of .906 and .813 respectively. The most commonly cited technique - in studies 44, 20 and 23 - was based on dynamic programming. Study 65 proposes techniques for generating a standardized corpus using Wikipedia. The corpus ‘Google Web Trillion Word Corpus’ in English was cited in study 44. There are other studies that present situations with specific corpus: hashtag splitting (45, 32 and 46) and domain splitting (33, 46 and 10).

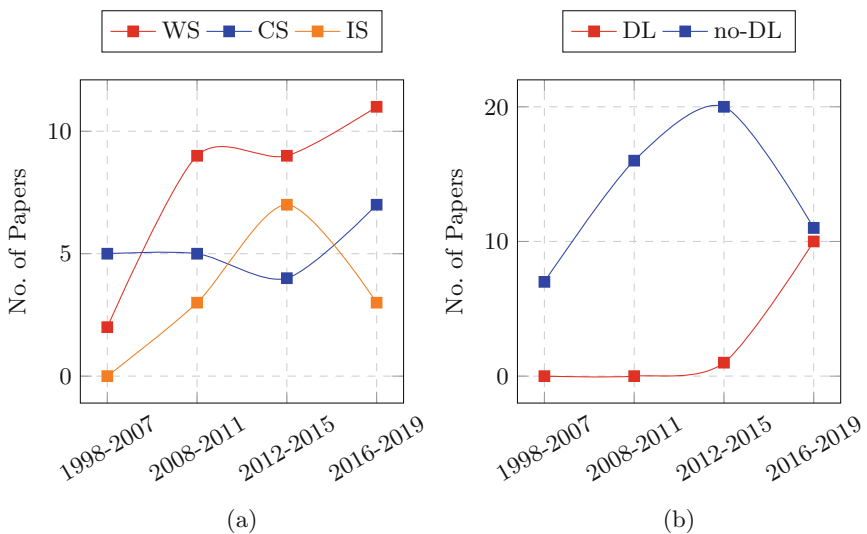


Fig. 3. At the left side (a) shows the use of WS, CS and IS from 1998 to 2019 and at the right side (b) shows the use of DL techniques from 1998 to 2019

4.3 RQ.3: What Is the State of the Art in WS in the Portuguese Language Context (PL)?

The authors in [5], developed a way of extract English compounds from the WordNet¹¹. The same approach could be used at Portuguese scenario, but we could not find any corpus annotations of compound words in the most recent WordNets¹². In order to know how many compound words exist in PL, we extracted 1804 words from a website¹³. Most of these words consisted of open

¹¹ <https://wordnet.princeton.edu>.

¹² <http://www.clul.ulisboa.pt/en/> and <http://wordnet.pt/>.

¹³ <http://www.linguabrasil.com.br/palavras-compostas.php>.

compounds (929), when a delimiter character separates the two parts of the word. According to the formal definition (Sect. 1), a problem consisting of a word with delimiter character does not characterize a WSP. In addition, compared with English and German, the number of closed compounds (without a delimiter character) in PL is much lower.

In this SLR, only 9 studies (1, 25, 37, 41, 45, 60, 62, 64 and 67) are considered universal, and only 2 (45 and 60) of them make direct reference to PL. Paper 45 refers to a specific application in hashtags and paper 60 is considered universal. In the studies, no software with direct support to the Portuguese language was found. All of them would need integration with specific training corpus in PL. Therefore, objective data for performance benchmarking are lacking. Considering this information, we can state that, compared to other languages, specific studies of WS for PL are lacking.

5 Discussion and Conclusions

In this SLR, we formally defined the problem of segmenting words written in the Latin Alphabet, present in many application domains and with different denominations. Several contexts were found and enumerated. The most relevant contexts are: word segmentation (WS), identifier splitting (IS) and compound splitting (CS) in natural language processing, software engineering and machine translation domains, respectively. We conducted a survey of techniques employed in each context, as well as a historical analysis of the use of deep learning techniques in recent years. Through data extraction and analysis, we conclude that, for each context, some specific techniques are more often than others. The most mature context in establishing a state of the art with standardized corpus is the IS. In the other contexts (CS and WS), there is no standard corpus.

References

1. Escartín, C.P.: Chasing the perfect splitter: a comparison of different compound splitting tools. In: LREC, pp. 3340–3347, May 2014
2. Garbe, W.: Fast Word Segmentation of Noisy Text (2018). <https://towardsdatascience.com/fast-word-segmentation-for-noisy-text-2c2c41f9e8da>
3. Kazakov, D., Manandhar, S.: Unsupervised learning of word segmentation rules with genetic algorithms and inductive logic programming. *Mach. Learn.* **43**(1–2), 121–162 (2001). <https://doi.org/10/fng8qb>, <https://www.scopus.com/inward/record.uri?eid=2-s2.0-0035312598&doi=10.1023%2FA%3a1007629103294&partnerID=40&md5=eaae5dc95f7c91cc97525afdf2bb2c17>, 144
4. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. EBSE Technical report 2, January 2007
5. Pedersen, T., Banerjee, S., Patwardhan, S.: compounds.pl - extract compound words (collocations) from WordNet - metacpan.org. <https://metacpan.org/pod/distribution/WordNet-Similarity/utls/compounds.pl>