Improving Chinese Dependency Parsing with Auto-extracted Dependency Triples

Likun Qiu[1,3], Lei Wu[2,3], Kai Zhao[3], Changjian Hu[3]

[1] School of Chinese Language and Literature, Ludong University, Yantai, China
[2]Institute of Automation, China Academy of Science
[3]NEC Laboratories, Beijing, China
qiulikun@pku.edu.cn

Abstract

To solve the data sparseness problem in dependency parsing, most previous studies used features extracted from large-scale auto-parsed data. Unlike previous work, we propose a novel approach to improve dependency parsing with dependency triples (DT) extracted by self-disambiguating patterns (SDP). The use of SDP makes it possible to avoid the dependency on a baseline parser and explore the influence of different types of DTs one by one. Experiments show that, when DT features are integrated into a maximum spanning tree (MST) dependency parser, the new parser improves significantly over the baseline MST parser. Comparative results also show that DTs with dependency relation labels perform much better than DTs without dependency relation label.

KEYWORDS

Dependency parsing; self-disambiguating pattern; raw corpus; annotated corpus; dependency triple

1 Introduction

To obtain dependency parsers with high accuracy, one promising direction is to use knowledge acquired from large-scale unannotated text, e.g., substructures extracted from auto-parsed data (i.e. "verb-object" and "modifier-head" structures in (Wu, 2003), case structure in (Yu, 2008) and subtrees in (Chen, 2009)).

However, since most of the substructures are extracted based on auto-parsed results, and the accuracies of auto-parsing is not very high (about 82% (Yu, 2008) on real POS-tag), there are many errors in the extracted substructures. Naturally, these errors might decrease the performance of dependency parsing. Moreover, most of previous researches (except Wu (2003)) used all kinds of substructures and took them as one type[1]. So it is difficult for us to know the influence of each type of substructures and hard to achieve further improvements.

Instead of extracting from auto-parsed data, we propose an approach to extract substructures directly from auto-segmented and auto-POS-tagged data. The approach is referred to as SDP-based approach. Here, SDP denotes self-disambiguating pattern, which

[1] Here, "type" denotes syntactic relation types such as coordinate, predicate-object, etc.

could resolve the ambiguity of a syntactic structure by itself. Since the precision of word segmentation and POS-tagging is much higher[2] than syntactic parsing and SDP could resolve syntactic ambiguity in a certain degree, the SDP-based method could extract labeled dependency triples effectively.

Moreover, unlike previous studies which improved performance by either using only verb-noun relations (Wu, 2003), or considering all kinds of syntactic collocations as only one pattern (Yu, 2008; Chen 2009), we propose to view dependency triples differently according to the grammatical relations.

To demonstrate the effectiveness of the proposed approach, we carried out experiments on Penn Chinese Tree Bank (CTB). The results showed that the proposed approach greatly improves the accuracy over the baseline parser and outperforms the state-of-art system.

The rest of this paper is organized as follows. Section 2 introduces related work. Section 3 describes how to extract DTs by using SDPs. Section 4 explains Chinese dependency parser using dependency triples. Experimental results are showed in section 5. Finally, a brief conclusion and our future work plan will be given in Section 6.

2 Related Work

To our knowledge, incorporating unannotated data into parsing model for dependency parsing is not new. Several previous studies relevant to this approach have been conducted.

Wu (2003) first used verb-noun relations to improve Chinese parsing and achieved good results. In the paper, predicate-object and attribute-head relations are involved. However, the verb-noun relations are only used in a rule-based parser to judge whether a given word pair is in a correct verb-noun relation. The paper proposed a promising approach. However, this rule-based approach hasn't been adopted in the following research especially in statistical dependency parsing. This paper follows the rule-based approach. It differs in that we use SDP-based method to extract DTs and so we don't need a baseline parser. And our work is focused on graph-based statistical parsing models.

Chen et al. (2008) proposed an approach that used the word pairs within two word distance for a transition-based parsing algorithm. Yu et al. (2008) constructed case structures from auto-parsed data and utilized both bi-lexical dependency and the parsing history of a head node in parsing models. Chen et al. (2009) used both bigram-subtrees and trigram-subtrees, which are also extracted from auto-parsed data. The approach in our paper differs in that we extract substructures (DT) by SDP-based method without using a baseline parser. Moreover, we first incorporate dependency relation type together with dependency direction into the substructure features.

Another relevant approach is to integrate word-pair classification model into dependency parsing. Jiang and Liu (2010) presented an intuitionistic method for dependency parsing, where a classifier is used to determine whether a pair of words forms a dependency edge. The word-pair classifier only used local context features. Our approach differs in that we haven't do word-pair classification directly. Instead, we first use SDP to resolve ambiguity of ambiguous structures and then incorporate frequency into features for dependency parsing. The word-pair classification work is done by the statistical parser itself.

3 SDP-based Approach for DT Extraction

In this paper, we would use SDPs to extract DTs from large scale unannotated data.

[2] Currently, the best performance of Chinese word segmentation has achieved about 96% on F-score, and the best accuracy of Chinese POS-tagging was 96.89% (Jin and Chen, 2008).

3.1 DT

A DT is a dependency triple with the form of {w_1, r, w_2}, in which r is the dependency relation between two words w_1 and w_2. In the sentence in Figure 1, the word 带来 (bring) is in four DTs: {发展 (fa-zhan, development), SUB, 带来 (dai-lai, bring)}, {将 (jiang, will), VMOD, 带来 (dai-lai, bring)}, {给 (gei, for), VMOD, 带来 (dailai, bring)}, {带来 (dai-lai, bring), OBJ, 机遇 (ji-yu, opportunity)}. After extracted from many different sentences, the relation type of a dependency triple is not dependent on its contexts.

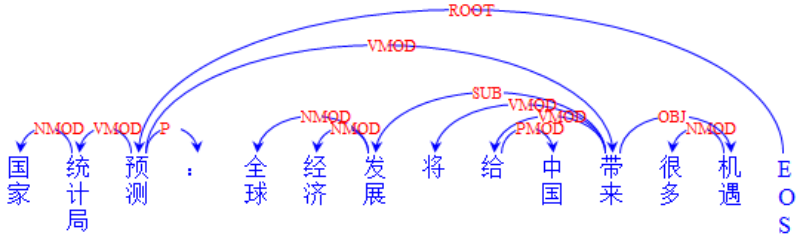

Figure 1: Example of Dependency Tree

3.2 SDP

There exist many ambiguous syntactic structures in natural languages. For instance, in Chinese, the dependency relation between a verb and a noun might be OBJ or NMOD; the relation between two verbs might be OBJ, NMOD, or VMOD.

However, in some cases, the relation of words can be decided without uncertainty. For example, although relation between a verb and a noun might be OBJ or NMOD, but in the case: "Verb +了+ Numeral + Quantifier + Anywords + Noun + EOS"[3], the relation of the verb and the noun is almost certainly OBJ. For example, in "收到 (shou-dao, receive)+了 (le, ~ed)+一(yi, one)+束(shu, bundle)+漂亮的(piao-liang-de, beautiful)+鲜花(xian-hua, flower)+EOS", the relation between "收到(shou-dao, receive)" and "鲜花(xian-hua, flower)" is OBJ. Such pattern is referred to as self-disambiguating pattern (SDP).

3.3 SDP-based Approach for DT Extraction

Since SDP can resolve the ambiguity of a structure, we can use it to extract correct DTs. For instance, given a SDP such as "Verb +了+ Numeral + Quantifier + Anywords + Noun + EOS", we might extract many instances of <Verb, OBJ, Noun>.

For each ambiguous structure, we might design one or more SDPs. Therefore, the number of SDPs in a certain language might be very large. In this paper, we present five SDPs to resolve four ambiguous structures, i.e. verb-object, subject-verb, quantifier-noun and attribute-head relation, [4] (see Table 1) in Chinese. In this table, the first column denotes the target DT that could be extracted by the SDP in the second column. Some high-frequency DTs of different dependency relations are show in Table 3, Table 2, Table 4 and Table 5 respectively. The samples in Table 3, Table 2 and Table 4 were extracted by the 1st,

[3] Here, "EOS" denotes the end of a sentence or clause; "Anywords" denotes any words occurring zero or many times; "[w]*" means the word w occurs zero or many times.

[4] The SUB and OBJ structure form the stretch of a common sentence; the noun-verb NMOD structure and verb-noun NMOD structure are the corresponding ambiguous structure of SUB and OBJ, respectively; the quantifier-noun NMOD structure is a typical relation of Chinese, where the quantifier and noun select mutually. That's why we choose them in our experiment.

2nd and 3rd SDP in Table 1 respectively. And samples in Table 5 were extracted by the 5th and 6th SDP in Table 1. The frequencies in these tables are counted on the Sogou Web Corpus (see detail in Section 5.1).

Note that each SDP in Table 1 only could extract part of instances of target DTs. For instance, the SDP "Verb +了+ Numeral + Quantifier + Anywords + Noun + EOS" means that it only can extract instances of <Verb, OBJ, Noun> in which the verb might occur before the word 了. In Chinese, many verbs such as 作为 can't collocate with 了. Therefore, this SDP can't extract instances including those verbs. More SDPs should be used to extract more complete instances.

Target DT	ID	SDP
<Noun, SUB, Verb>	1	BOS + [Adjective | Noun | Pronoun | Numeral | Quantifier |的]* + Noun + Adverb + Verb + 了 + [Adjective | Noun | Pronoun| Numeral | Quantifier |的]* + Noun + EOS
<Verb, OBJ, Noun>	2	Verb +了+ Numeral + Quantifier + Anywords + Noun + EOS
<Quantifier, NMOD, Noun>	3	了+ Numeral + NominalQuantifier + Anywords + Noun + EOS
<Noun, NMOD, Verb>	4	的 + Noun + Verb + EOS
<Verb, NMOD, Noun>	5	了+ Numeral + NominalQuantifier + Anywords +Verb + Noun + EOS
	6	的 + Verb + Noun + EOS

Table 1. SDPs used in this paper

Subject	Verb	Frequency
他(ta, he)	表示(biao-shi, express)	9359
版权(ban-quan, copyright)	属于(shu-yu, belong to)	5835
投资者(tou-zi-zhe, investor)	了解(liao-jie, understand)	5358
他(ta, he)	指出(zhi-chu, claim)	4010
人士(ren-shi, person)	表示(biao-shi, express)	3462
记者(ji-zhe, reporter)	采访(cai-fang, interview)	3402
数据(shu-ju, data)	显示(xian-shi, show)	3095

Table 2. DT Samples of Subject-Predicate Relation

Verb	Noun	Frequency
解决(jie-jue, solve)	问题(wen-ti, problem)	64965
提出(ti-chu, raise)	要求(yao-qiu, claim)	54313
发挥(fa-hui, play)	作用(zuo-yong, role)	46052
取得(qu-de, score)	成绩(cheng-ji, achievement)	42867
奠定(dian-ding, lay)	基础(ji-chu, foundation)	37495
积累(ji-lei, accumulate)	经验(jing-yang, experience)	34762

采取(cai-qu, take)	措施(cuo-shi, measure)	33441

Table 3. DT Samples of Predicate-Object Relation

Quantifier	Noun	Frequency
(一 (yi, one))个(ge, a)	百分点(bai-fen-dian, percentage)	11919
(一 (yi, one))个(ge, a)	月(yue, month)	4127
(一(yi, one))个(ge, a)	问题(wen-ti, question)	2983
(一(yi, one))句(ju, a)	话(hua, word)	2729
(一(yi, one))段(duan, a)	时间(shi-jian, period)	2725
(一(yi, one))件(jian, a)	事(shi, thing)	2562
(一(yi, one))系列(xi-lie, a series of)	措施(cuo-shi, measure)	2432

Table 4. DT Samples of Quantifier-Noun Relation

Attribute	Head	Frequency
合作(he-zuo, cooperation)	关系(guan-xi, relation)	23283
合作(he-zuo, cooperation)	协议(xie-yi, agreement)	16567
发展(fa-zhan, development)	机遇(ji-yu, opportunity)	8597
工作(gong-zuo, work)	会议(hui-yi, conference)	8368
发展(fa-zhan, growth)	空间(kong-jian, space)	7798
管理(guan-li, management)	制度(zhi-du, system)	7305
解决(jie-jue, solve)	方案(fang-an, solution)	6105

Table 5. DT Samples of Attribute-Head Relation

4 Chinese Dependency Parser Using DT Features

We generate new features based on the extracted DTs and refer them as DT-based features. Since these features only contain two words, they correspond to the first-order features in the MST parsing model. Second-order or higher-order features would not be tried in this paper.

A DT-based feature is represented as follows:

$$feature(w_i,w_j)=id_{i,j}\text{-}type_{i,j}\text{-}direction_{i,j}$$

where $id_{i,j}$, $type_{i,j}$ and $direction_{i,j}$ denote the frequency, dependency relation type and dependency direction of the DT respectively.

All the extracted DTs are grouped into different sets in terms of frequencies. With experiments and reference to (Chen, 2009), we chose the following way. DTs are grouped into three sets: "high-frequency (HF)", "middle-frequency (MF)" and "low-frequency (LF)". HF, MF and LF are used as set IDs for the three sets respectively. The following are the settings: if the frequency of a DT is larger than the threshold λ_1, it is in set HF; else if the frequency is larger than λ_2, it is in set MF; else it is in set LF. We store the set ID for every DT in L_{st}.

The dependency type set contains three elements: SUB, OBJ and NMOD. The dependency directions of SUB, OBJ and NMOD are "left", "right" and "left" respectively. Here, "left" means in the DT the left word depends on the right word. If a DT with SUB label is matched, the value of "direction" would be set as "left".

For instance, if the frequency of <解决 (jie-jue, solve), OBJ, 问题 (wen-ti, problem)> is larger than λ_1, its set ID is HF. Then the system would generate a feature of HF-OBJ-Right.

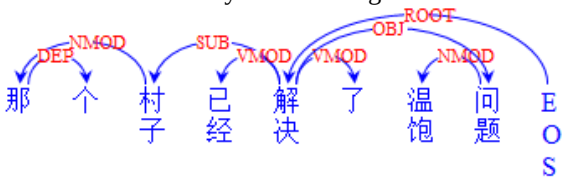

Figure 2 Example of Dependency Tree

Figure 3 shows all the features generated from the sentence in Figure 2.

Feature{个,NMOD,村子} = MF-NMOD-Left
Feature{村子,SBJ,解决} = LF-SBJ-Left
Feature{解决,OBJ,问题} = HF-OBJ-Right
Feature{解决,OBJ,温饱} = HF-OBJ-Right

Figure 3 Examples of DT features

5 Experiments

5.1 Experimental Setting

We made experiments on the Penn Chinese Treebank (CTB)5.0[5] (Xue et al., 2002). CTB is converted into dependency structures using a standard set of head rules by the tool "Penn2Malt"[6] (Yamada and Matsumoto, 2003). In CTB 5.0, Section 1-270 and 400-931 are used for training, Section 271-300 for testing, and Section 301-325 for development. Data partition and POS-tags on CTB 5.0 are the same as the settings in Chen (2008, 2009) and Yu (2008). All the evaluation metrics are calculated on the dependency relations, in which the modifier is not punctuation.

Two unannotated corpus are used for extracting DTs. The first one is a self-made corpus, called Raw-Corpus. Raw-Corpus contains about 20M sentences, which are collected from Chinese news websites from January to December 2006. We use the second order MSTParser[7] as our baseline parser. It is trained with Section 1-270 and 400-931 of CTB 5.0. All the sentences in the Raw-Corpus are parsed by the baseline parser for extracting DTs from auto-parsed sentences directly.

The second corpus is Sogou Web Corpus[8] V2.0. This corpus contains 120G web pages, yet we only used the previous 30G for extracting DTs by SDP-based method.

The PKU Contemporary Chinese Corpus (Yu, 2002) is used as the annotated corpus for experiments. This Corpus, which is segmented and POS-tagged manually, contains all the news articles of the People's Daily newspaper published in China in the year of 1998 and 2000.

[5] http://www.cis.upenn.edu/~chinese/
[6] http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html
[7] http://mstparser.sourceforge.net
[8] http://www.sogou.com/labs/dl/t.html

We use the same feature with McDonald and Pereira (2006) and adopt the default settings of MSTParser throughout the paper: iters=10; training-k=1; decode-type=proj; order=2. When grouping DTs by frequencies, the parameters λ_1 and λ_2 are set as 100 and 10 respectively.

5.2 Experimental Results

5.2.1 Results of DT Extraction

The results of DT extraction are show in Table 6. There are 55,175 and 1485 dependency triples (SUB, OBJ, NMOD) in training set and test set respectively. Auto-parsed DTs are extracted from auto-parsed result (using the baseline parser) on the Chinese Raw-Corpus. SDP-based DTs are extracted from Sogou Corpus based on the five proposed SDPs and using ICTCLAS2009 (Zhang, 2002)[9] as the segmentation and POS-tagging tool.

Since the dependency triples in training set and test set are context-dependent yet DT is context-free, it is difficult for us to evaluate DT in terms of precision. We only evaluate the coverage rate of DT, i.e. the percentage of DTs in training set and test set that have been covered by the given DT set. Table 6 shows that auto-parsed DTs can cover more than SDP-based DTs. It is mainly because we only used five SDPs and can only cover a small part of the SUB, OBJ and NMOD DTs. The usefulness of the three DT sets would be evaluated by dependency parsing evaluation.

DT Source	Quantity	Coverage Rate
Training Set	55,175	-
Test Set	1485	-
Auto-parsed	8.52M	53.9
SDP-based	1.69M	16.1

Table 6: DT Quantity

5.2.2 Results of Dependency Parsing

The dependency parsing results of proposed parsers are show in Table 7. Five parsers are compared together with a baseline parser:

- Baseline: We use the second order MSTParser[10] as our baseline parser.
- Auto1: The parser which uses DTs extracted from auto-parsed data, without dependency label.
- Auto2: The parser which uses DTs extracted from auto-parsed data, with dependency label.
- SDP1: The parser which only uses seed DTs without dependency label. That is, all kinds of DT are considered as one kind.
- Proposed (SDP2): The parser which uses seed DTs with different dependency label.

Note that all the SDP1/2 and Auto1/2 parsers only used DTs of SBJ, OBJ and NMOD relations.

Table 7 shows that all the five parsers outperform the baseline parsers. There is an absolute improvement of 1.26 points (UAS) by adding DT-based features in the proposed parser. The improvement of parsing with DT-based features is significant in McNemar's Test ($p < 10^{-5}$). Figure 4 and Figure 5 show the dependency trees of the same sentences

[9] http://ictclas.org/index.html
[10] http://mstparser.sourceforge.net

created by the baseline parser and the proposed parser, respectively. After using the DT <会谈 (hui-tan, interview), SUB, 具有 (ju-you, have)>, the correct subject of 具有 (ju-you, have) was found by the proposed parser.

The comparative results between SDP/Auto1 and SDP/Auto2 parsers show that integrating dependency relation type into features is very useful for dependency parsing.

The comparative results between SDP1/2 and Auto1/2 parser show that SDP-based method could extract DTs effectively. Note that SDP1/2 parser, which used five SDPs and only cover about 16% dependency triples in CTB 5.0, outperform the parser which used all DTs from auto-parsed data and cover 53.9% dependency triples.

Parser	UAS(%)	LAS(%)
Baseline	88.21	87.19
Auto1	88.71 (+0.50)	87.71(+0.52)
SDP1	88.72 (+0.51)	87.71(+0.52)
Auto2	89.19 (+0.98)	88.14(+0.95)
Proposed (SDP2)	89.25 (+**1.04**)	88.21(+**1.02**)

Table 7: Dependency parsing results for proposed parsers

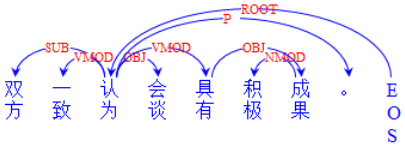

Figure 4: Result created by the baseline parser

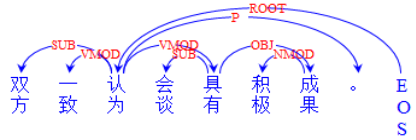

Figure 5: Result created by the proposed parser

5.2.3 Comparative Results of Dependency Parsing

Table 8 shows the comparative results, where Chen08 refers to the parser of Chen et al. (2008), Yu08 refers to the parser of Yu et al. (2008), Chen09b/s refers to the parsers of Chen et al. (2009). Specially, Chen09b only used bigram features yet chen09s used both bigram and trigram features. Chen et al. (2009) is the best reported results for this data set. The results show that our proposed parser outperforms previous best results that used bigram features (chen092b). Note that the experiment of chen092b used collocations (without dependency relation label) extracted from auto-parsed corpus, which has been segmented and POS-tagged manually before being parsed automatically (the PKU Corpus of 1998).

We also test the performance of using DTs extracted from the corpus that has been segmented and POS-tagged manually before being parsed automatically. In this experiment, we parsed the PKU Corpus of 1998 and 2000 respectively using the baseline parser and then extracted DTs from the auto-parsed corpus. The following experimental setup was the same as the proposed parser. That is, DT-based features were constructed as in Section 4. Table 9 shows the results. Seen from Table 8 and Table 9, we may find that our experiment

is obviously better than that of Chen092b. This result shows the effectiveness of the proposed feature construction method, which combine dependency label into features.

Parser	UAS(%)	LAS(%)
Baseline	88.21	87.19
Chen08	86.52	-
Yu08	87.26	-
Chen092b	89.16	-
Chen092s	89.43	-
Proposed	89.25	88.21

Table 8: Dependency parsing results for the proposed parsers and for previous work

Corpus	UAS(%)	LAS(%)
1998	89.53	88.47
2000	89.44	88.43
1998 & 2000	89.69	88.63

Table 9: Dependency Parsing Results of Using Auto-extracted DTs from Word-segmented and POS-tagged Corpus

6 Conclusion and Future Work

We presented an effective approach to improve dependency parsing using DTs extracted by SDP-based method. The experiments showed that the SDP-based method could improve dependency parsing significantly. Experimental results also showed that DTs with dependency labels perform much better than that without dependency label.

Much work can be done to further exploit SDP-based method. For example, we only used five SDPs in this paper, which can only cover a small part of the dependency triples. We would design more SDPs to deal with dependency relations such as adverbial-head, predicate-complement and coordinate structure.

Acknowledgement

We would like to thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Natural Science Foundation of China (No. 61103089 and No. 90920011) and National High Technology Research and Development Program of China (863 Program) (No. 2012AA011101).

7 References

W. Chen, D. Kawahara, K. Uchimoto, Y. Zhang, and H. Ishara. 2008. Dependency parsing with short dependency relations in unlabeled data. In *Proceedings of IJCNLP 2008*.

W. Chen, J. Kazama, K. Uchimoto, and K. Torisawa. 2009. Improving Dependence Parsing with Subtrees from Auto-Parsed Data. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 570-579. Singapore, 6-7 August 2009.

W. Jiang, and Qun Liu. 2010. Dependency Parsing and Projection Based on Word-pair Classification. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. pages 12-20. Uppsala, Sweden, 11-16 July 2010.

G. Jin and X. Chen. 2008. The Fourth International Chinese Language Processing Bakeoff: Chinese Word Segmentation, Named Entity Recognition and Chinese Pos Tagging. In *Proceedings of the 6th SIGHAN Workshop on Chinese Language Processing*, pages 69-81.

R. McDonald, and F. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of EACL 2006*.

D. Wang, X. Tu, X. Zheng and Z. Tong. 2008. Collocation Extraction with Multiple Hybrid Strategies. *Journal of Tsinghua University (Science & Technology)*, Vol. 48, No. 4, pages 608-612.

A. Wu. 2003. Learning Verb-Noun Relations to Improve Parsing. In *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing*. pages.119-124.

N. Xue, F. Chiou, and M. Palmer. 2002. Building a Large-Scale Annotated Chinese Corpus. In *Proceedings of the 18th International Conference on Computational Linguistics.*

K. Yu, D. Kawahara, and S. Kurohashi. 2008. Chinese Dependency Parsing with Large Scale Automatically Constructed Case Structures. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1049-1056. Manchester, August 2008.

S. Yu, H. Duan, X. Zhu, and B. Swen. 2002. The basic processing of Contemporary Chinese Corpus at Peking University. In *Journal of Chinese Information Processing*, Vol. 16, No. 5, pp. 49-64.