INTEGRATED TECHNIQUES FOR PHRASE EXTRACTION FROM SPEECH

Marie Meteer
J. Robin Rohlicek

BBN Systems and Technologies
Cambridge, Massachusetts 02138
mmeteer@bbn.com
rohlicek@bbn.com

ABSTRACT

We present an integrated approach to speech and natural language processing which uses a single parser to create training for a statistical speech recognition component and for interpreting recognized text. On the speech recognition side, our innovation is the use of a statistical model combining N-gram and context-free grammars. On the natural language side, our innovation is the integration of parsing and semantic interpretation to build references for only targeted phrase types. In both components, a semantic grammar and partial parsing facilitate robust processing of the targeted portions of a domain. This integrated approach introduces as much linguistic structure and prior statistical information as is available while maintaining a robust full-coverage statistical language model for recognition. In addition, our approach facilitates both the direct detection of linguistic constituents within the speech recognition algorithms and the creation of semantic interpretations of the recognized phrases.

1. INTRODUCTION

Language modeling for speech recognition has focused on robustness, using statistical techniques such as n-grams, whereas work in language understanding and information extraction has relied more on rule based techniques to leverage linguistic and domain information. However, the knowledge needed in these two components of a speech language system is actually very similar. In our work, we take an integrated approach, which uses a single grammar for both language modeling and language understanding for targeted portions of the domain and uses a single parser for both training the language model and extracting information from the output of the recognizer.

The goal of our work is provide speech recognition capabilities that are analogous to those of information extraction systems: given large amounts of (often low quality) speech, selectively interpret particular kinds of information. For example, in the air traffic control domain, we want to determine the flight IDs, headings, and altitudes of the planes, and to ignore other information, such as weather and ground movement.

The following is a summary of the main techniques we use in our approach:

- *Integration of N-gram and context free grammars for speech recognition*: While statistically based Markov-chain language models (N-gram models) have been shown to be effective for speech recognition, there is, in general, more structure present in natural language than N-gram models can capture. Linguistically based approaches that use statistics to provide probabilities for word sequences that are accepted by a grammar typically require a full coverage grammar, and therefore are only useful for constrained sublanguages. In the work presented here, we combine linguistic structure in the form of a partial-coverage phrase structure grammar with statistical N-gram techniques. The result is a robust statistical grammar which explicitly incorporates syntactic and semantic structure. A second feature of our approach is that we are able to determine which portions of the text were recognized by the phrase grammars, allowing us to isolate these phrases for more processing, thus reducing the overall time needed for interpretation.

- *Partial parsing*: It is well recognized that full coverage grammars for even subsets of natural language are beyond the state of the art, since text is inevitably errorful and new words frequently occur. There is currently a upsurge in research in partial parsing in the natural language community (e.g., Hindle 1983, Weischedel, et al. 1991), where rather than building a single syntactic tree for each sentence, a forest is returned, and phrases outside the coverage of the grammar and unknown words are systematically ignored. We are using the partial parser "Sparser" (McDonald 1992), which was developed for extracting information from open text, such as Wall Street Journal articles.

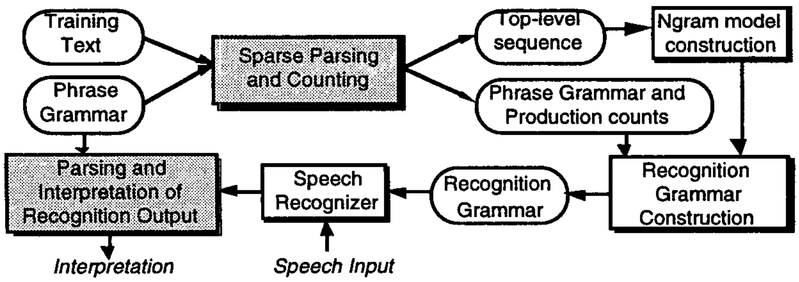
Training Text
Phrase Grammar
Sparse Parsing and Counting
Top-level sequence
Ngram model construction
Phrase Grammar and Production counts
Parsing and Interpretation of Recognition Output
Speech Recognizer
Recognition Grammar
Recognition Grammar Construction
Interpretation
Speech Input

Figure 1: Overall Approach

- *Semantic grammar:* Central to our approach is the use of a minimal, semantically based grammar. This allows us to build targeted grammars specific to the domain. It also makes the grammar much more closely tied to the lexicon, since the lexical items appear in the rules directly and in general there are many categories, each covering only a small number of lexical items. As Schabes (1992) points out in reference to lexicalized stochastic tree adjoining grammars (SLTAG), an effective linguistic model must capture both lexical and hierarchical information. Context free grammars using only syntactic information fail to capture lexical information.

Figure 1 shows a block diagram of the overall approach with the two components which use the parser shaded: the model construction component and the interpretation component.

For both the language modeling and information extraction, we are using the partial parser Sparser (McDonald 1992). Sparser is a bottom-up chart parser which uses a semantic phrase structure grammar (i.e. the nonterminals are semantic categories, such as HEADING or FLIGHT-ID, rather than traditional syntactic categories, such as CLAUSE or NOUN-PHRASE). Sparser makes no assumption that the chart will be complete, i.e. that a top level category will cover all of the input, or even that all terminals will be covered by categories, effectively allowing unknown words to be ignored. Rather it simply builds constituent structure for those phrases that are in its grammar.

In Section Two, we describe language modeling, and in Three, we focus on semantic interpretation. In Section Four, we present the results of our initial tests in the air traffic control domain, and finally we conclude with future directions for the work.

2. LANGUAGE MODELING

There are two main inputs to the model construction portion of the system: a transcribed speech training set and a phrase-structure grammar. The phrase-structure grammar is used to partially parse the training text. The output of this is: (1) a top-level version of the original text with subsequences of words replaced by the non-terminals that accept those subsequences; and (2) a set of parse trees for the instances of those nonterminals.

3.1 Rule Probabilities

Figure two below shows a sample of the rules in the ATC grammar followed by examples of transcribed text and the text modified by the parser. Note that in this case, where goal is to model aircraft identifiers and a small set of air traffic control commands, other phrases like the identification of the controller, traffic information, etc., are ignored. They will be modelled by the n-gram, rather than as specific phrases.

R1 (def-rule land-action > ("land"))
R2 (def-rule takeoff-action > ("takeoff"))
R3 (def-rule takeoff-action > ("go"))
R4 (def-rule clrd/land > ("cleared" "to" land-action)
R5 (def-rule clrd/takeoff > ("cleared" "to" takeoff-action))
R6 (def-rule clrd/takeoff > ("cleared" "for" takeoff-action)))
R7 (def-rule tower-clearance > (runway clrd/land)
R8 (def-rule tower-clearance > (runway clrd/takeoff))

Figure 2: Phrase structure rules for tower clearance

>Nera twenty one zero nine runway two two right cleared for takeoff
>COMMERCIAL-AIRPLANE TOWER-CLEARANCE

>Nera thirty seven twelve Boston tower runway two two right cleared for takeoff
>COMMERCIAL-AIRPLANE Boston tower TOWER-CLEARANCE

>Jet Link thirty eight sixteen Boston tower runway two two right cleared for takeoff traffic on a five mile final landing two two right
>COMMERCIAL-AIRPLANE Boston tower TOWER-CLEARANCE traffic on a five mile final landing RUNWAY

>Jet Link thirty eight zero five runway two two right cleared for takeoff sorry for the delay
>COMMERCIAL-AIRPLANE TOWER-CLEARANCE sorry for the delay

Figure 3: Training text modified by parser

Using the modified training text we construct a probabilistic model for sequences of words and non-terminals. The parse trees are used to obtain statistics for the estimation of production probabilities for the rules in the grammar. Since we assume that the production probabilities depend on their context, a simple count is insufficient. Smoothed maximum likelihood production probabilities are estimated based on context dependent counts. The context is defined as the sequence of rules and positions on the right-hand sides of these rules leading from the root of the parse tree to the non-terminal at the leaf. The probability of a parse therefore takes into account that the expansion of a category may depend on its parents. However, it does not take into consideration the expansion of the sister nonterminals, though we are currently exploring means of doing this (cf. Mark, et al. 1992).

In the above grammar (Figure 2), the expansion of TAKEOFF-ACTION may be different depending on whether it is part of rule 5 or rule 6. Therefore, the "context" of a production is a sequence of rules and positions that have been used up to that point, where the "position" is where in the RHS of the rule the nonterminal is. For example, in the parse shown below (Figure 4), the context of R2 (TAKEOFF-ACTION > "takeoff") is rule 8/position 2, rule 6/position 3. We discuss the probabilities required to evaluate the probability of a parse in the next section.

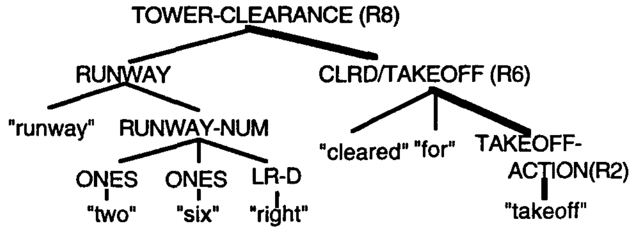

Figure 4: Parse tree with path highlighted

In order to use a phrase-structure grammar directly in a time-synchronous recognition algorithm, it is necessary to construct a finite-state network representation If there is no recursion in the grammar, then this procedure is straightforward: for each rule, each possible context corresponds to a separate subnetwork. The subnetworks for different rules are nested. We are currently comparing methods of allowing limited recursion (e.g. following Pereira & Wright 1990). Figure 5 shows the expansion of the rules in from Figure 2.

There have been several attempts to use probability estimates with context free grammars. The most common technique is using the Inside-Outside algorithm (e.g. Pereira & Schabes 1992, Mark, et al. 1992) to infer a grammar over bracketed texts or to obtain Maximum-Likelihood estimates for a highly ambiguous grammar. However, most require a full coverage grammar , whereas we assume that only a selective portion of the text will be covered by the grammar. A second difference is that they use a syntactic grammar, which results in the parse being highly ambiguous (thus requiring the use of the Inside-Outside algorithm). We use a semantic grammar, with which there is rarely multiple interpretations for a single utterance.

3.2 Probability Estimation

Both the context-dependent production probabilities of the phrase grammar and one for the Markov chain probabilities for the top-level N-gram model must be estimated. We use the same type of "backing-off" approach in both cases. For the phrase grammar, we estimate probabilities of the form

$$P(r_{n+1} \mid (r_1, p_1), (r_2, p_2), ..., (r_n, p_n))$$

where r_i are the rules and p_i are the positions within the rules. In the N-gram case, we are estimating

$$P(s_{n+1} \mid s_1, s_2, ..., s_n)$$

where $s_1, s_2, ..., s_n$ is the sequence of words and non-terminals leading up to s_{n+1}. In both cases, the estimate is based on a combination of the Maximum-Likelihood estimate, and the estimates in a reduced context:

$$P(r_{n+1} \mid (r_2, p_2), ..., (r_n, p_n))$$

and

$$P(s_{n+1} \mid s_2, ..., s_n).$$

The Maximum-Likelihood (ML) estimate reduces to a simple relative-frequency computation in the N-gram case. In the phrase grammar case, we assume that the parses are in general unambiguous, which has been the case so far in our domain. Specifically, we only consider a single parse and accumulate relative frequency statistics for the various contexts in order to obtain the ML production

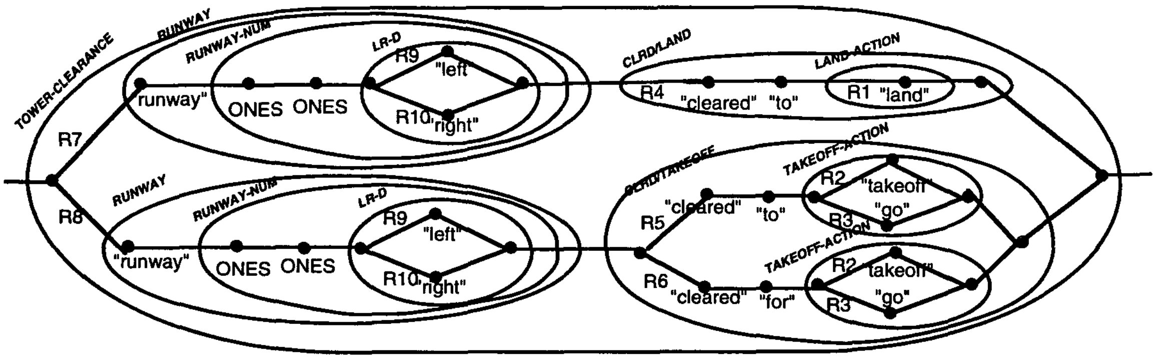

Figure 5: Finite state network

probabilities.

The approach we use to backing off is described in Placeway, et al. (1993). Specifically, we form

$$P^{BO}(y \mid x_1,..., x_n) = P^{ML}(y \mid x_1, .., x_n)\ (1 - \theta) + P^{BO}(y \mid x_2, ..., x_n)\ \theta.$$

The value of θ depends on the context $x_1, ..., x_n$ and is motivated by approximation of the probability of

$$\theta = r / (n + r)$$

where r is the number of different next symbols/rules seen in the context and n is the number of times the context was observed.

3. INFORMATION EXTRACTION

The final stage of processing is the interpretation of the recognized word sequence. We use the same phrase structure grammar for interpretation as that used to build the recognition model. However, in this last phase, we take advantage of the semantic interpretation facility of the parser.

Most approaches to natural language understanding separate parsing (finding a structural description) from interpretation (finding a semantic analysis). In the work presented here, we use a single component for both. The Sparser system integrates parsing and interpretation to determine "referents" for phrases incrementally as they are recognized, rather than waiting for the entire parse to finish. The referent of a phrase is the object in the domain model that the phrase refers to. For example, the initial domain model consists of objects that have been created for entities which are known to the system in advance, such as airlines. When the name of an airline is recognized, such as "Delta", its referent is the airline object, #<airline delta>. Referents for entities that cannot be anticipated, such as number sequences and individual airplanes, are created incrementally when the phrase is recognized. Figure 6 shows an example of the edges created by the parser and their referents.

When a referent actually refers to an entity in the world, such as a runway or airplane, then the same referent object is cataloged and reused each time that entity is mentioned. The referent for a number sequence is a number object with the value the sequence represents. The referent for the entire phrase "Delta three five nine" is an object of type airplane. In some cases, the object will also be indexed by various subparts (such as indexing a flight ID by the digit portion of the ID) to aid in disambiguating incomplete subsequent references. For example, in the pilot reply in Figure 6, indexing allows the system to recognize that the number "three five nine" actually refers to the previously mentioned Delta flight.

We extend the notion of referent from simply things in the world to utterance acts as well, such as commands. Each time a command is uttered, a new referent is created. Command referents are templates which are created when some core part is recognized and then added to compositional as other (generally optional) information is recognized. So following our earlier example of tower clearances, rules 4, 5, and 6 instantiate a takeoff clearance template and fill in the action type, whereas rules 7 and 8 fill in the "runway" field. We show examples of each of these groups and the templates in Figure 7 below:

```
R6  (def-rule clrd/takeoff ("cleared" "for" takeoff-action)
     :referent (:function create-tower-clearance third))
R8  (def-rule tower-clearance (runway clrd/takeoff)
     :referent (:function add-to-tower-clearance second first))

#<tower-clearance
     Type: TOWER-CLEARANCE
     ACTION: #<TAKEOFF>
     RUNWAY: #<Runway 26L>>
```

Figure 7: Rules with referents and completed template.

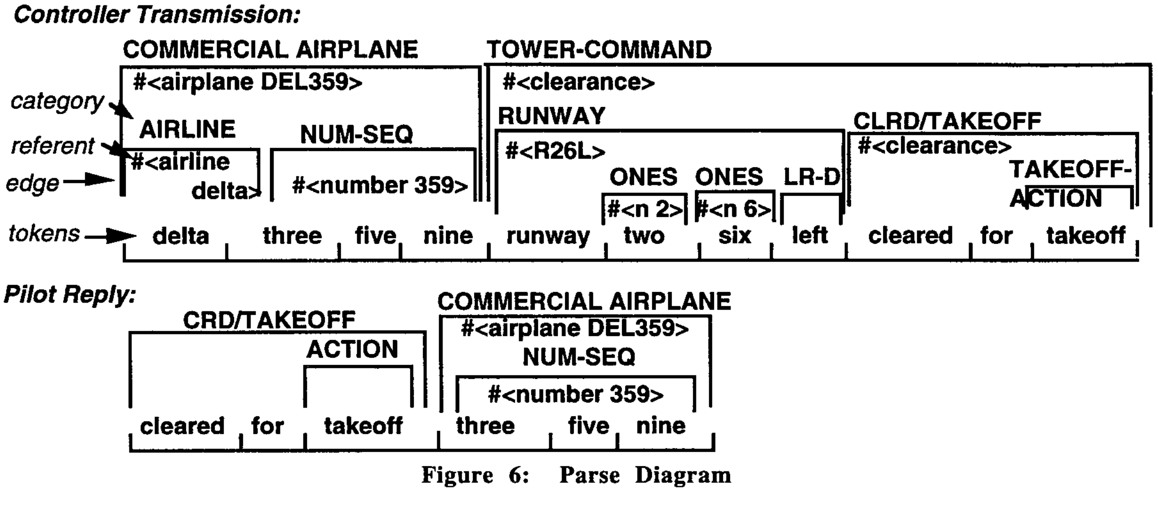

Figure 6: Parse Diagram

4. RESULTS

This approach was first applied in the Gisting system (Rohlicek, et al. 1992), where the goal was to extract flight IDs from off-the-air recordings of ATC communications. In this application, the input is extremely noisy and recognition performance is generally quite poor. We report the general word recognition accuracy and flight ID recognition accuracy for both the combined phrase structure and n-gram language models (as described in section 2), and just n-grams. The training data consists of 2090 transcribed controller transmissions. Testing data consists of 469 transmissions of average length 16. The results are presented for controller transmissions where the start and end times of the transmissions are known.

As shown in Table 1, the overall word accuracy was improved only slightly (70% to 72%), which was expected since we only modeled a small portion of the domain. However, the best result was in the fraction of flight IDs detected, where we halved the miss rate (from 11% down to 5%).

	Word Recognition				FID rec. accuracy
	Sub.	Del.	Ins	Acc.	
N-gram & phrase	18.6	4.5	5.2	72	57
N-gram	20.4	5.0	4.3	70	53

Table 1: Results for Gisting experiment.

The next set of experiments we ran focused on comparing general word accuracy with word accuracy in the targeted portion of the domain (i.e. that portion covered by the grammar). Using a different ATC dataset (still operational data, but recorded in the tower rather than off the air), we compared bi-grams with our combined rule based and n-gram approach The grammar covered approximately 68% of the training data. We tested not only the overall word accuracy, but also the word accuracy in those portions of the text that were modeled by the grammar.

	Bi-gram		Integrated	
	words correct	word error	words correct	word error
Overall word accuracy	64.3	45.9	68.2	40.4
Word accuracy in phrases	58.6	46.0	74.8	36.2

Table 2: Comparison between Bi-grams and integrated approach.

As shown in Table 2, not only was there an improvement in the overall word score using the integrated vs. the bi-gram language model, we can see that the improvement in accuracy in the targeted portion of the domain was much greater in the integrated approach.

Our third set of experiments focused on the information extraction portion of the system. We evaluated the ability of the parser to extract two kinds of commands from the output of recognition. In these experiments, we took truth to be the performance of the parser on the transcribed text, since we did not have truth annotated for these phrases in our test data. (It has been our experience in working with flight IDs, which were annotated, that in the ATC domain the phrases are regular enough that the parser will extract nearly 100% of the information in the targeted categories. The errors that occur are generally caused by restarts, speech errors, or transcription errors.)

Using the same training and test conditions as the first set of experiments described above[1], we extracted phrases for tower clearances using the grammar partially shown above (Figure 2), and direction orders, which generally consisted of a direction to turn and some heading. The test set consisted of 223 controller utterances and we scored as correct only exact matches, where the same referent object was found and all of the fields matched exactly. Results are shown in Table Three.

Exact Match	**Direction Order**	**Tower Clearance**
Total in reference	38	118
Total in recog.	35	117
Precision	91.4%	43.6%
Recall	81.6%	43.2%
False Positives	1	11
Misses	5	12
Errors	2	7
Partial Match		
Precision		64.4
Recall		63.8

Table 3: Precision and recall in extracting information.

We observe that the precision and recall for direction orders is drastically better than that for tower clearances, even though the grammars for the two are very similar in size. One difference, which we would like to explore further, is

[1] Note on difference was that these tests were done on recognition results after automatic segmentation and classification according to pilot and controller, which generally decrease recognition accuracy.

that the direction orders grammar was part of the language model which was used for recognition, whereas tower clearances were not modelled by the phrase grammar, only the n-gram. To know if this was a factor, we need to compare the actual word recognition accuracy for these two phrase types.

In looking at the results for tower clearances, we found that although the exact match score was very low, there were many partial matches, where for example the runway and or the action type (takeoff, land, etc.) were found correctly, even though the entire tower clearance was not recognized. In order to take into account these partial matches, we rescored the precision and recall, counting each individual piece of information (runway, action, and clearance), so that an exact match gets a score of 3 and partial matches score a 1 or 2. Using this measure, we got a significantly improved performance: precision 64.4 and recall 63.8. These results highlight one of the main the advantage of this approach, that even with errorful input, useful information can be found.

FUTURE WORK

We have shown the the approach described here both improves overall word accuracy in recognition and provides a means for extracting targeted information even recognition performance is quite poor. Our next goal is to apply the technique to new domains. As part of this effort we are developing a set of tools for building and evaluating grammars.

We are also also applying these techniques in new applications. In particular, we have recently performed experiments in Event Spotting, which is an extension of wordspotting where the goal is to determine the location of phrases, rather than single keywords. We used the parser/extraction portion of the system to find examples of phrase types in the corpus and to evaluate the results, as well as in the language model of the recognizer. In an experiment detecting time and date phrases in the Switchboard corpus (which is conversational telephone quality data), we saw an increase in detection rate over strictly bi-gram or phoneme loop language models (Jeanrenaud, et al. 1994).

Acknowledgement This work was funded by ARPA and the Air Force Rome Laboratory under contract F30602-89-C-0170.

REFERENCES

Hindle, Don (1983) "Deterministic Parsing of Syntactic Non-fluencies" *Proc. of the 21st Annual Meeting of the Association for Computational Linguistics*, June 15-17, pp. 123-128.

Jeanrenaud, P., Siu, M., Rohlicek, R., M., Meteer, Gish, H. (1994) "Spotting Events in Continuous Speech" to appear in *Proceedings of International Conference of Acoustics, Speech, and Signal Processing* (ICASSP), April 1994, Adelaide, Australia.

Mark, K., Miller, M., Grenander, U., & Abney, S. (1992) "Parameter Estimation for Constrained Context-free Language Models" in *Proceedings of the Speech and Natural Language Workshop*, February, 1992, Morgan Kaufmann, San Mateo, CA, p. 146-149.

McDonald, David D. (1992) "An Efficient Chart-based Algorithm for Partial Parsing of Unrestricted Texts" in *Proceedings of the 3rd Conference on Applied Natural Language Processing*, April 1-3, 1992, Trento, Italy, pp.193-200.

Pereira, F. & Schabes, E. (1992) "Inside-Outside Reestimation from Partially Bracketed Corpora" in Proceedings of the Speech and Natural Language Workshop, February, 1992, Morgan Kaufmann, San Mateo, CA, p. 122-127.

Pereira, F. & Wright, R. (1991) "Finite-state approximation of phrase structured grammars" *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, June 18-21, 1991, Berkeley, California, pp.246-255.

Placeway, P., Schwartz, S., Fung, P., & Nguyen, L., (1993) “Estimation of Powerful Language Models from Small and Large Corpora”, in *Proceedings of International Conference of Acoustics, Speech, and Signal Processing* (ICASSP).

Rohlicek, R., Ayuso, D., Bates, M., Bobrow, R., Boulanger, A., Gish, H., Jeanrenaud, M., Meteer, M., Siu, M. (1992) "Gisting Conversational Speech" in *Proceedings of International Conference of Acoustics, Speech, and Signal Processing* (ICASSP), Mar. 23-26, 1992, Vol.2, pp. 113-116.

Schabes, E. (1992) "Stochastic Tree-Adjoining Grammars" in *Proceedings of the Speech and Natural Language Workshop*, February, 1992, Morgan Kaufmann, San Mateo, CA, p. 140-145.

Weischedel, R., Ayuso, D., Bobrow, R., Boisen, S., Ingria, R., Palmucci, J. (1991) "Partial Parsing: A report on work in progress" in *Proceedings of the Speech and Natural Language Workshop*, February, Morgan Kaufmann, Pacific Grove, CA, p. 204-209.