Temporal Relation Classification in Hebrew

Guy Yanko
Efi Arazi School
of Computer Science
Reichman University
guyy1232@gmail.com

Shahaf Pariente
Efi Arazi School
of Computer Science
Reichman University
shahafp2@gmail.com

Kfir Bar
Efi Arazi School
of Computer Science
Reichman University
kfirbarmail@gmail.com

Abstract

Temporal Relation Classification (TRC) is a fundamental task in natural language processing (NLP) and is essential for achieving a comprehensive understanding of a natural language. Given a document containing two event mentions, the objective of this task is to discern which of the two events happened first. Existing TRC datasets predominantly consist of texts written in English. To accommodate the growing interest in relevant NLP applications for Hebrew, we introduce a new TRC dataset for Hebrew. Professional annotators labeled Hebrew documents with TRC labels, adhering to guidelines adapted from a similar project on English and with some changes required to address some unique aspects of the Hebrew language. Overall, we annotated a corpus of 28,757 words, corresponding to 7,260 pairs of events. In addition to releasing the new dataset, which can be accessed at https://github.com/shahafp/TRC-Hebrew, we train several baseline models for TRC and report their performance.

1 Introduction

Events in a story are not necessarily mentioned in a chronological order. Constructing a timeline of events mentioned in a document is crucial for comprehending the primary narrative of the story. The timeline, also known as the chronology of events, can be applied to various use cases. For instance, clinicians may utilize the timeline to conveniently explore their patients' disease progression. Moreover, a model can follow text-based instructions more accurately with a timeline in place. Building a timeline comprises several subtasks, with the two primary subtasks being: 1) Event detection, which involves identifying the most important events in a given text; and 2) Temporal relation classification (TRC), which entails determining the correct order of two given events. Consider the following Hebrew text as an example:

"אכלתי את התפוח אחרי ששטפתי אותו"
"I ate the apple after washing it"

The event detection subtask is about detecting only the relevant events for our domain of interest; in this example: אכלתי "I ate" and שטפתי "I washed". In this case, both words are verbs representing actions which we subsequently mark as events. In the TRC subtask, we classify each pair of events using a closed set of labels to establish their chronological order. In this example, the events אכלתי "I ate" and שטפתי "I washed" should be assigned the label AFTER to indicate that "I ate" occurs after "I washed" in chronological order.

This study focuses on creating a Hebrew dataset for the second subtask, TRC, which is generally defined as a classification problem involving two events accompanied by their contextual information. Utilizing the annotated data, we train multiple TRC baseline models using all available Hebrew language models.

2 Related Works

To the best of our knowledge, there is no publicly available Hebrew dataset for TRC. TRC in English, however, has been a subject of considerable focus within the scientific literature.

Ning et al. (2018b) proposed a multi-axis annotation scheme for TRC, improving inter-annotator agreement (IAA), ascending from the conventional 60's to a high of 80's, as measured on the Cohen's Kappa scale. This new annotation scheme enabled the use of crowdsourcing, thereby mitigating the laborious demands on individual annotators. In this work, the authors introduced the most dominant TRC dataset in English, MATRES, which contains news documents manually annotated with TRC labels. Overall, MATRES

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023, pages 261–267

contains 12,736 training instances and 837 test instances. We derive inspiration from their annotation guidelines and adapt them to Hebrew. According to their guidelines, all verbs mentioned in a document are selected as events. Every pair of events (n, m) is manually labeled with one of four labels reflecting the chronological order between the two. The four labels are: BEFORE (n occurs before m), AFTER (n occurs after m), EQUAL (n and m occur simultaneously), and VAGUE (it is impossible to determine which event preceded the other). We follow the same guidelines. Previous works on building TRC datasets include TempEval (Verhagen et al., 2007), TimeBankDense (Chambers et al., 2014), RED (OGorman et al., 2016), and TCR (Ning et al., 2018a).

Various computational strategies have been employed to train a TRC model for English. Most of them use language models (Han et al., 2021, 2019a,b), some form of a global inference mechanism (Zhou et al., 2021; Ning et al., 2019; Mathur et al., 2021), and certain linguistic information extracted from the text (Zhang et al., 2021; Wang et al., 2022b,a). The best model for English TRC has been reported recently by (Zhou et al., 2022), who extracted relational syntactic and semantic structures, and encoded them using a graph neural network achieving 84% F1-score on MATRES.

3 TRC in Hebrew

We annotated articles from the Universal Dependencies Corpus for Hebrew (Tsarfaty, 2013; McDonald et al., 2013),[1] containing about 500 articles, corresponding to 6,143 sentences. We choose to annotate this dataset since it already contains part-of-speech labels for each word, simplifying the task of identifying all verbs as actions.

3.1 Annotation Preparation

To convert the treebank articles into TRC samples, we first identify all events mentioned in each document. Events are defined as any word labeled as a verb, excluding infinitive verbs and gerunds due to their lack of temporal meaning in Hebrew. Next, we create two-sentence sliding windows that run through every article. For instance, an article with three sentences yields two windows, the first spanning the first and second sentences, and the second covering the second and third sentences. For every window, we generate one sample for every pair of events mentioned within the window text. The two events are marked by enclosing them within bracketed special tokens. For example:

[א1] זכיתי [/א1] בלוטו אז [א2] החלטתי [/א2]
לקנות רכב חדש

In other words, each sample is a copy of the entire window text, with the two specific events marked with brackets. Overall, we process 78 articles from the treebank, corresponding to 7,260 TRC samples, which we distribute among the annotators.

3.2 Annotation Process

We hired two annotators, both possessing academic backgrounds, with one individual having expertise in linguistics. The dataset was curated by two of the authors. We start with a set of annotation pilots to validate and improve our annotation guidelines. We use INCEpTION.[2] The annotators were provided with a set of instructions to assist in determining the appropriate label for each pair of events (n, m). Inspired by the method used in MATRES, we define two guiding questions that can assist the annotators in choosing the correct label:

1. `Q1`: Could it be possible for the start time of n to precede the start time of m?
2. `Q2`: Could it be possible for the start time of m to precede the start time of n?

From these two binary questions, there are four possible answer combinations, each of which maps to a specific label, according to Table 1.

Answer to `Q1`	Answer to `Q2`	Label
Yes	No	BEFORE
No	Yes	AFTER
No	No	EQUAL
Yes	Yes	VAGUE

Table 1: Label assignment.

Generally, our annotation guidelines derive from those established by MATRES, with some modifications suitable for Hebrew. For more information, please refer to Appendix C. Table 2 summarizes the label distribution of the annotated dataset. We compared the label distribution with that of the original MATRES dataset. The two distributions appear quite similar, except that in our

[1] https://github.com/UniversalDependencies/UD_Hebrew-HTB

[2] https://inception-project.github.io/

dataset, there are more EQUAL labels and fewer VAGUE labels. Some examples from the dataset can be found in Appendix D.

Label	Count	Percentage
BEFORE	3,092	42% (49%)
AFTER	2,374	33% (23%)
EQUAL	618	9% (2%)
VAGUE	1,176	16% (26%)

Table 2: Label distribution (Total: 7,260). In parentheses: For comparison, the label distribution of the original MATRES dataset is provided.

3.3 Multi-Axis Annotations

Events can have temporal relationships only if they share the same axis. Time axes are defined following the method outlined in (Ning et al., 2018b). The guidelines allow for the assignment of each event in a given text to one of three types of axes: main, parallel, and orthogonal. The main axis represents the primary sequence of events in the text, and there exists only one such axis. On the other hand, parallel axis (may be more than one) represents a secondary timeline axis in which events occur on a distinct timeline from the main or another parallel axis. Typically, events that occur in quotation or that form part of a hypothesis are assigned to a parallel axis. Events on a parallel axis can only have temporal relationships with other events on the same axis.

An orthogonal axis, of which there may be more than one in a given story, denotes an alternate timeline originating from an event that belongs to another axis, also known as the "connecting" event. In other words, an orthogonal axis is a branch that stems from the original axis. Generally speaking, a pair of events can have a non-VAGUE temporal relationship only if they belong to the same axis. A connecting event is capable of having a non-VAGUE relationship with events transpiring on the two axes it links together.

The temporal relationship between two events that do not share the same axis is VAGUE. During the annotation pilots, we learned that in most cases, annotators can intuitively understand the concept of using multiple time axes. However, when confronted with complex situations, it was necessary to establish some guidelines for identifying the axes. Our guidelines are released with the dataset.

3.4 Inter-Annotator Agreement

After completing two pilot annotation projects, in which we made suitable modifications to the guidelines, we carried out two rounds of annotations. In the first round, all the TRC samples were annotated by at least one annotator, and about 55% were annotated by both annotators to measure consistency and agreement. Following this, we had a detailed discussion with the annotators to clarify any ambiguities in the guidelines. Additionally, we identified the samples that showed disagreement between the two annotators (1,073 windows out of 3,998 windows which were annotated by both annotators) and conducted a second annotation round for these specific samples. After the second round concluded, the authors adjudicated any remaining discrepancies in the annotations.

To measure the inter-annotator agreement (IAA) level, we calculate Cohen's kappa for each round of annotation separately as well as for both rounds combined. In addition to the standard IAA calculations, following common evaluation practices on MATRES (Ning et al., 2018b), we define a relaxed version of IAA. If one annotator labels a sample as VAGUE, and the other as either AFTER or BEFORE, it is not necessarily a complete disagreement, given that VAGUE inherently encompasses both temporal directions. Therefore, in the relaxed IAA we exclude disagreements regarding VAGUE. Table 3 summarizes all IAA values.

Method	1st Round	2st Round	Combined
Standard	0.62	0.53	0.81
Relaxed	0.84	0.77	0.91

Table 3: Cohen's Kappa values for each annotation round and both rounds combined.

4 TRC Models

We fine-tune a number of models for Hebrew TRC based on some existing Hebrew foundation language models. The model's input consists of a sentence containing the two events. Inspired by Soares et al. (2019) we mark the events with distinct special tokens, as shown in Section 3.1. We use Hebrew marker; specifically, we employ [א1] to denote the beginning of the first event and [א1/] to denote its ending. The second event is marked similarly, but replacing 1 with 2.[3]

[3]We experimented with alternative marker types, and the results indicate that this approach is optimal. Performance

Model	Architecture					
	SEQ-CLS		ESS		EMP	
	M/ avg	W/ avg	M/ avg	W/ avg	M/ avg	W/ avg
AlephBERTGimmel	0.62	0.74	0.65	**0.78**	0.65	**0.78**
HeBERT	0.56	0.65	0.56	0.64	0.59	**0.75**
mBERT	0.48	0.64	0.61	**0.74**	0.62	0.73
AlephBERT	0.64	0.75	0.67	**0.77**	0.67	**0.77**

Table 4: Relaxed F1 scores. W/ avg = Weighted Average F1; M/ avg = Macro Average F1.

We use four Hebrew language models: AlephBERT (Seker et al., 2022), HeBERT (Chriqui and Yahav, 2022), mBERT cased (bert-base-multilingual) (Devlin et al., 2019), and AlephBERTGimmel (Guetta et al., 2022), all having 110M parameters, obtained directly from Hugging Face's transformer library. Inspired by Soares et al. (2019), we experiment with three sequence classification architectures:

Sequence Classification (SEQ-CLS). We use the [CLS] vector as input for the classification layer.

Event Start State (ESS). We concatenate the output vector of the start markers of both events, and use it for classification.

Event Mention Pooling (EMP). For each event, we apply maxpooling on the output vectors of all its word pieces, excluding markers. The two pooled vectors are concatenated into a single vector, serving as the classification layer's input.

The classification layer is always implemented as a simple linear layer. See Appendix A for training details.

4.1 Models Evaluation

We use the macro and weighted average F1 metrics. As previously stated, we adopt the relaxed F1 score metric (ignoring mistakes of non-VAGUE predictions on VAGUE samples). In order to train and evaluate our models, we divide the dataset into training and evaluation sets using an 80/20 split. We ensure that each article is assigned to only one of the sets. The split is released with the dataset. Table 4 summarizes the relaxed F1 scores measured on the evaluation set. The corresponding non-relaxed scores can be found in Appendix B. The best performance is achieved by AlephBERTGimmel using ESS and EMP. To study the relationship between the training-set size and the model performance, we train our AlephBERTGimmel/ESS architecture using a growing size of the training set, and monitor its performance. We train the model using 15%, 25%, 50%, 75%, and 100% of the training set. For each size, we train the model three times using different seeds for training-set sampling. Figure 1 shows the mean of the macro-average F1 scores over the three executions, with a standard deviation of 0.02 for the 15% portion and 0.01 for the rest of the portions. The results show an increasing improvement with the expansion of the training set. However, the improvement becomes less significant after using about 50% of the data, implying that our newly created Hebrew TRC dataset provides an adequate volume of data for constructing robust models.

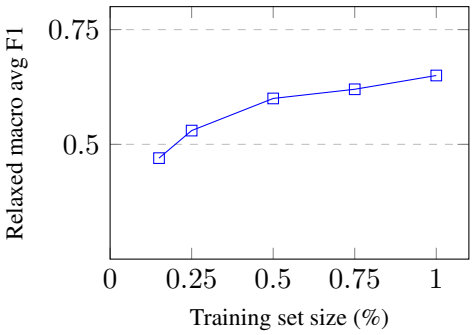

Figure 1: Training on different training set sizes.

5 Conclusion

In this study, we present a Hebrew TRC dataset, employing TRC annotation techniques that have been previously validated. The dataset contains 7,260 samples, manually annotated by professional annotators, following guidelines developed as part of this work. While our dataset currently only includes news articles, we plan to add documents from other domains. We fine-tune a range of TRC models under various classification settings

diminished when markers were not used.

and evaluate their performance. We have made the dataset and models accessible for research purposes, under a license (CC BY-NC-SA 4.0).[4]

6 Limitations

There are three main limitations to this work, two are related to the dataset itself and the third to the model we provide along with the dataset. First, we use a single, domain-specific source of news articles. While we provide full annotation guidelines, a new domain might require some adjustments. The second limitation is related to VAGUE annotations. Our guidelines label any two events in a window that don't share the same axis as VAGUE. However, a different label could be chosen for such a pair to distinguish it from a pair of events on the same axis, but with both directive questions answered with Yes. The third limitation pertains to the model itself. We have proposed an evaluation methodology termed "relaxed". In this approach, predictions that do not align with the VAGUE label are considered true positives, even when the actual label is VAGUE. This deviation stems from the inherent complexities and variability associated with the VAGUE label.

References

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Avihay Chriqui and Inbal Yahav. 2022. HeBERT & HebEMO: a Hebrew BERT model and a tool for polarity analysis and emotion recognition. *INFORMS Journal on Data Science*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics.

Eylon Guetta, Avi Shmidman, Shaltiel Shmidman, Cheyn Shmuel Shmidman, Joshua Guedalia, Moshe Koppel, Dan Bareket, Amit Seker, and Reut Tsarfaty. 2022. Large pre-trained models with extra-large vocabularies: A contrastive analysis of Hebrew BERT models and a new one to outperform them all.

Rujun Han, I Hsu, Mu Yang, Aram Galstyan, Ralph Weischedel, Nanyun Peng, et al. 2019a. Deep structured neural network for event temporal relation extraction. *arXiv preprint arXiv:1909.10094*.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019b. Joint event and temporal relation extraction with shared representations and structured prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 434–444, Hong Kong, China. Association for Computational Linguistics.

Rujun Han, Xiang Ren, and Nanyun Peng. 2021. ECONET: Effective continual pretraining of language models for event temporal reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Punta Cana, Dominican Republic.

Puneet Mathur, Rajiv Jain, Franck Dernoncourt, Vlad Morariu, Quan Hung Tran, and Dinesh Manocha. 2021. TIMERS: Document-level temporal relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 524–533, Online. Association for Computational Linguistics.

Ryan T McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith B Hall, Slav Petrov, Hao Zhang, Oscar Täckström, et al. 2013. Universal dependency annotation for multilingual parsing. In *Proc. of ACL*.

Qiang Ning, Zhili Feng, Hao Wu, and Dan Roth. 2018a. Joint reasoning for temporal and causal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2278–2288, Melbourne, Australia. Association for Computational Linguistics.

Qiang Ning, Sanjay Subramanian, and Dan Roth. 2019. An improved neural baseline for temporal relation extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6203–6209, Hong Kong, China. Association for Computational Linguistics.

Qiang Ning, Hao Wu, and Dan Roth. 2018b. A multi-axis annotation scheme for event temporal relations. In *ACL*.

Tim OGorman, Kristin Wright-Bettner, and Martha Palmer. 2016. Richer event description: Integrating event coreference with temporal, causal and bridging annotation. In *Proceedings of the 2nd Workshop on Computing News Storylines (CNS 2016)*, pages 47–56.

Amit Seker, Elron Bandel, Dan Bareket, Idan Brusilovsky, Refael Greenfeld, and Reut Tsarfaty. 2022. AlephBERT: Language model pre-training and evaluation from sub-word to sentence level. In

[4] https://github.com/shahafp/TRC-Hebrew

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 46–56, Dublin, Ireland. Association for Computational Linguistics.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. *CoRR*, abs/1906.03158.

Reut Tsarfaty. 2013. A unified morpho-syntactic scheme of stanford dependencies. In *Proc. of ACL*.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. Semeval-2007 task 15: Tempeval temporal relation identification. In *Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*, pages 75–80.

Haoyu Wang, Hongming Zhang, Yuqian Deng, Jacob R Gardner, Muhao Chen, and Dan Roth. 2022a. Extracting or guessing? improving faithfulness of event temporal relation extraction. *arXiv preprint arXiv:2210.04992*.

Liang Wang, Peifeng Li, and Sheng Xu. 2022b. DCT-centered temporal relation extraction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2087–2097.

Shuaicheng Zhang, Lifu Huang, and Qiang Ning. 2021. Extracting temporal event relation with syntactic-guided temporal graph transformer. *arXiv preprint arXiv:2104.09570*.

Jie Zhou, Shenpo Dong, Hongkui Tu, Xiaodong Wang, and Yong Dou. 2022. RSGT: Relational structure guided temporal relation extraction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2001–2010, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yichao Zhou, Yu Yan, Rujun Han, J. Harry Caufield, Kai-Wei Chang, Yizhou Sun, Peipei Ping, and Wei Wang. 2021. Clinical temporal relation extraction with probabilistic soft logic regularization and global inference. In *AAAI*.

A Training Information

We fine-tune all our models for the duration of 10 epochs, using a learning value of 3×10^{-5} and batch-size value of 32. To train our models, we used one Nvidia Tesla T4 with 16GB of memory. It takes about 45 minutes to train each model. The model cards from Hugging Face we use are: AlephBERTGimmel: `imvladikon/alephbertgimmel-base-512`, HeBERT: `avichr/heBERT`, mBERT: `bert-base-multilingual-cased`, and AlephBERT: `onlplab/alephbert-base`.

B Non-Relaxed F1 Scores

Table 5 summarizes the non-relaxed F1 scores of our experiments with the different models, as reported in the text.

C Adjustments for Hebrew TRC

We have made some adjustments to the MATRES annotation guidelines of English to accommodate the unique challenges of Hebrew. Two primary additions to the guidelines are:

1. Verbal nouns and infinitive verbs are not counted as events.

2. Auxiliary verbs (in Hebrew פעלי עזר, such as הלך, היה) are not labeled as part of the events. For example, in the text: "היה מתאמן" "he was practicing", only מתאמן is labeled as an event.

For more information, please refer to the annotation guidelines, provided in as supplementary data.

D Examples

In Table 6 we provide some samples from the dataset, along with their gold labels. We provide one full annotated document (in json format) in the supplementary data.

Model	Architecture					
	SEQ_CLS		ESS		EMP	
	M/ avg	W/ avg	M/ avg	W/ avg	M/ avg	W/ avg
AlephBERTGimmel	0.48	0.58	0.50	0.60	0.49	0.59
HeBERT	0.32	0.44	0.32	0.44	0.44	0.56
mBERT	0.33	0.44	0.43	0.54	0.46	0.56
AlephBERT	0.47	0.58	0.48	0.59	0.50	0.60

Table 5: **Non-relaxed** F1 scores. M/ avg = Macro Average F1; W/ avg = Weighted Average F1.

Sample	Label
במחנה פליטים סמוך לשכם נפגע תושב מקומי מפגיעת רימון גז מדמיע, שנורה בידי חיילים. מקורות פלשתיניים [א1] **מסרו** [/א1] כי חיילים, שירדו ממיניבוס לבן בשכונת ראפידיה בשכם, [א2] **ניפצו** [/א2] בעזרת אבנים זגוגיות של מכוניות חונות של תושבים מקומיים, ביניהן גם את חלונות מכוניתו של דר מוסטפא מקבול.	AFTER
האיש שנפל קורבן לכדורי אקדח שנורו על-ידי ערבי, [א1] **הציתו** [/א1] במותו את האיבה לערבים. האש הזאת [א2] **תוסיף** [/א2] ללחוש, גם אם הלהבה של יום קבורתו של מאיר כהנא תדעך במקצת.	BEFORE
התחרות היתה בהרצליה. קפספקו מתל אביב [א1] **ניצח** [/א1] במירוץ אופניים שהיה סביב כיכר המדינה בתל אביב, לאחר [א2] **שעבר** [/א2] 8 ק"ם ב-41.	AFTER
אמנם נוצחנו פעם אחת, אבל לא שיחקנו נגד הגדולות. אם [א1] **ננצח** [/א1] באחד משלושת המשחקים הבאים נגד הפועל תל אביב, מכבי תל אביב והפועל גליל עליון זה יהיה הישג אדיר, [א2] **שיתן** [/א2] לנו דחיפה קדימה".	BEFORE
עדיין לא הוגש כתב הגנה. 48 דיירים [א1] **המתגוררים** [/א1] ברחוב החלמונית 3, 5 ו7 בקריית ראשון בראשון-לציון [א2] **הגישו** [/א2] השבוע סדרה של תביעות נגד סולל בונה.	BEFORE
משה ויזל, מנכ"ל החברה, זומן לבירורים בהתאחדות חברות הביטוח וכן עם מנהלי חברות ביטוח. [א1] **טענו** [/א1] בפניו שהוא [א2] **כורת** [/א2] את הענף של הביטוח עליו הוא יושב.	AFTER
דובר מחלקת המדינה של ארה"ב, אמר אתמול כי ארה"ב "היתה בקשר הדוק עם ישראל במשך כל התקופה" שמאז תחילת המשבר במפרץ. הוא [א1] **חזר** [/א1] [א2] **ושיבח** [/א2] את גישת ה"פרופיל הנמוך" שישראל נוקטת.	EQUAL
כך [א1] **עשה** [/א1] גם בליגה להגנה בברוקלין. בארץ [א2] **שיחרר** [/א2] בלמים אצל קבוצות השוליים, והצליח להניח על השולחן רעיונות שגם אנשים מן היישוב הוגים בהם, אך אינם מעיזים לבטא אותם (עד לעת האחרונה).	VAGUE

Table 6: Examples from the dataset.