Automatic Recognition of Narrative Drama units: a structured learning approach

Danilo Croce
Dept. of Enterprise Engineering
University of Roma, Tor Vergata (Italy)
croce@info.uniroma2.it

Roberto Basili
Dept. of Enterprise Engineering
University of Roma, Tor Vergata (Italy)
basili@info.uniroma2.it

Vincenzo Lombardo
Dept. of Informatics,
University of Torino, (Italy)
vincenzo.lombardo@unito.it

Eleonora Ceccaldi
DIBRIS
University of Genova, (Italy)
eleonoraceccaldi@gmail.com

Abstract

Drama is a story told through the live actions of characters; dramatic writing is characterized by aspects that are central to identify, interpret, and relate the different elements of a story. The Drammar ontology has been proposed to represent core dramatic qualities of a dramatic text, namely Actions, Agents, Scenes and Conflicts, evoked by individual text units. The automatic identification of such elements in a drama is the first step in the recognition of their evolution, both at coarse and fine grain text level. In this paper, we address the issue of segmentation, that is, the partition of the drama into meaningful unit sequences We study the role of editorial as well as content–based text properties, without relying on deep ontological relations. We propose a generative inductive machine learning framework, combining Hidden Markov models and SVM and discuss the role of event information (thus involving agents and actions) at the lexical and grammatical level.

1 Introduction

Drama is a story told through the live actions of characters. The Drammar ontology [LDP18, DLPar] identifies the core dramatic qualities of a dramatic text, namely Actions, Agents, Units/Scenes, and Conflicts, implicitly evoked by the dramatic text, as claimed by the scientific literature on drama analysis.

Drama relies on an internal coherence and a rich set of eventualities, related to the interactions among characters and the insurgence and resolution of conflicts. Dramas are very well structured. As a running example, we address the incipit of Anton Chekhov's "*The Cherry Orchard*", in its English translation [Che17]:

A room which is still called the nursery. One of the doors leads into ANYA'S room. It is close on sunrise. ... DUNYASHA comes in with a candle, and LOPAKHIN with a book in his hand.

LOPAKHIN. The train's arrived, thank God. What's the time?

In: A. Jorge, R. Campos, A. Jatowt, S. Bhatia (eds.): Proceedings of the Text2StoryIR'19 Workshop, Cologne, Germany, 14-April-2019, published at http://ceur-ws.org

DUNYASHA. It will soon be two. [*Blows out candle*] It is light already.

Schematically:

- individual utterances are denoted by the correspondent acting characters;
- some editorial notes (in italics) interleave with the spoken parts where the authors suggest the environment changes or specific happenings;
- a strict separation between spoken and editorial fragments is imposed.

Our current research objective aims at supporting the automatic annotation of a drama, able to outline the evolution of the dramatic elements above through discrete events.

1.1 Events in narrative texts

Following the observer point of view, [SZR07] propose the following definitions for the events:

- An EVENT is "a segment of time at a given location, that is conceived by an observer to have a beginning and an end; granularity of events can go from a second or less to tens of minutes".
- An EVENT MODEL is "an actively maintained representation of the current event, which is updated at perceptual event boundaries".
- The EVENT SEGMENTATION is "the perceptual and cognitive process by which a continuous activity is segmented into meaningful events".

From the psychological literature, we have that readers structure a narrative text into a series of events in order to understand and remember the text (cf. the experiments of [ZT01, ZS07, ZSSM10]). Events are coded at clause level. Relevant information for the narrative coding includes, e.g. [SZR07]:

- TIME and SPACE information (as the presence of spatial changes, e.g., moving from one room to another inside a house can be meaningful);
- OBJECTS, given the interaction of characters with elements of a scene;
- change of CHARACTER, revealed by the changes of the subject of a clause;
- CAUSES (causal relationship over activities) and GOALS (new goal-directed activities), to be coded as core dimensions of EVENTS.

Usually, clauses are also coded for terminal punctuation (e.g., periods and question marks) and non terminal punctuation (e.g., commas and semicolons). As the annotation of such dramatic aspects is time consuming, we aim at automatizing it, relying upon the lexical, grammatical, and editorial information, expressed by individual clauses. In this way, events can be recognized and properly segmented along the dramatic text. We will refer hereafter this process as the *event segmentation*.

1.2 Event Segmentation: Related Work

Event segmentation is a task traditionally tackled in NLP according to sentence boundary detection methods (e.g. [Hea94], [SDDK11]) or cohesion-based clustering models (e.g. [Cho00]). Text segmentation methods usually search for the set of segments in a text that optimize some form of coherence of the content. Word usage is modeled in TextTiling [Hea94] for each sentence in a sequence and the two sides of a potential boundary are selected when large lexical difference is found. Prosodic features and lexical features are taken into account to model discourse as in the Hidden Markov Model segmentation proposed in [YXX$^+$16]. The lexical connectivity strength between two adjacent fragments of a text is used as hint in DivSeg ([SDDK11]). Unsupervised approaches are based on *probabilistic models* (e.g. [Hea94], C99 [Cho00] or the DotPlotting algorithm [Rey94]) or *agglomerative clustering* [Yaa99]). In the former group, terms frequencies are used to identify topical segments (dense dot clouds on the graphic). e.g. DotPlotting [Rey94]. In the latter group, dendrograms are induced over paragraphs and transformed into a hierarchical segmentation [Yaa99]. *Lexical chains methods* are applied in an unsupervised manner as they exploit semantic lexicons to model word associations and semantic relations. In these methods, a chain links multiple occurrences of a term in the document: it is considered broken when there are too many sentences between two occurrences of a term. The Segmenter system ([KKM98]) detects such broken points across a document according to possibly multiple chains. Some of the methods use lexical resources or forms of ontological similarity to model similarity metrics between text blocks (sentences or paragraphs), based on semantic information (e.g. recognized named entity in the text). Wordnet or Wikipedia-based methods have been proposed to define semantic similarity metrics between text units. Recently, deep learning methods have been applied to Text Segmentation, specifically to the Topic-based segmentation task. In particular [LSJ18] presents an end-to-end segmentation model: first, a bidirectional recurrent neural network is used to encode input text sequences, and then, another recurrent neural network is used together with a pointer network to select text boundaries in the input sequence. Although very appealing, since it does not require hand-crafted features definition, this method requires a significant amount of training material, made of several hundreds of annotated documents.

2 A structured learning approach to drama segmentation

In line with most of the above event segmentation approaches, we will rely on a machine learning perspective by assuming a set of textual resources as the triggering observations:

- L is a set of fully annotated drama fragments, whose segments are completely known (e.g., the example units 4 to 6 in Checkov's *The Cherry Orchard* reported in the paper Appendix or the nunnery scene from Shakespeare's Hamlet[LPD16], respectively);

- O_L is a very small scale corpus, made of fragments from the possibly partially annotated opera (e.g., the complete annotated dramas "The Cherry Orchard" or "Hamlet", respectively, though they are usually neither segmented, nor annotated);

- $O_A(L)$ is a large scale corpus of unannotated texts of the same author (e.g., all of Checkov's plays or Shakespeare's plays, respectively);

- $O_E(A(L))$ is a comprehensive corpus of the drama works of the same epoch (e.g., Contemporary play or Elizabethan theatre plays, respectively).

So, we rely on the chain $L \subset O_L \subset O_A(L) \subset O_E(A(L))$.

We propose an integration of unsupervised and supervised learning processes acting: our first attempt is to use the comprehensive $O_E(A(L))$ to generate a lexical resource focused on the work and author style: according to unsupervised methods, such as [MCCD13], we can rely on word embedding for a large scale dictionary of lexical items: these generalize lexical semantics within the underlying targeted text genre. The proposal is to inject this information into the supervised steps that address the labeled material L, in order to fully label the entire work O_L in an accurate manner. Annotated examples in L are the basic source of information for the segmentation stage.

Hereafter, we concentrate on the variety of lexical, grammatical and aspectual features (e.g. the mode and transitivity of a number of verbs involved in the dramatic action), suitably exploited for training a sequence labeling component over O_L. We propose a structured learning paradigm based on independent kernels for training SVMs over L ([STC04]) and apply them within a Markovian modeling, isomorphic to HMM. The major steps are thus:

- (**PreTraining**) Use $O_E(A(L))$ to acquire lexical information in the form of a neural language model (in line with [MCCD13]), expressing general semantic properties of individual words. A specific treatment of some classes of words is here applied. For example, character names (e.g. Dunyasha and Lopakhin in *The Cherry Orchard* or Hamlet and Ophelia in *Hamlet*). This is a standard a-priori information for a drama that is mapped into the category label CHARACTER, in order to minimize sparsity.

- (**Feature Modeling and Extraction**) Feature extraction is applied to derive textual, editorial and narrative features, as discussed in Section 2.2

- (**Model Optimization**) Then, a structured Machine Learning model is applied to achieve segmentation as an IOB-like sentence labeling process, in order to organize sentences in units and hierarchies of scenes. The adopted algorithm is known as SVM-HMM ([TJHA05], adopted in [CB11, BCV$^+$16]).

2.1 A Markovian Support Vector Machine

The aim of a Markovian formulation of SVM is to make the classification of a input example $x_i \in \mathbb{R}^n$ (belonging to a sequence of examples) dependent on the labels assigned to the previous elements in a history of length m, i.e., $x_{i-m}, \ldots, x_{i-1}$. In our classification task, a drama is a sequence of utterances $\mathbf{x} = (x_1, \ldots, x_s)$, each of them representing the example x_i, i.e., the specific i-th paragraph. Given the corresponding sequence of expected labels $\mathbf{y} = (y_1, \ldots, y_s)$, a sequence of m step-specific labels (from a dictionary of d symbols) can be retrieved, in the form $y_{i-m}, \ldots, y_{i-1}$. In our machine learning setting, labels are related to the Segmentation task: we will thus adopt the `IOB` notation so that each element in the drama will be associated to the label `B` if it is at the `B`eginning of a UNIT, `I` if it is `I`nside it, `O` if it is `O`ut of the UNIT itself. In order to make the classification of x_i also dependent on the previous decisions, we augment the feature vector of x_i by introducing a projection function $\psi_m(x_i) \in \mathbb{R}^{md}$ that associates each example with a $md-$dimensional feature vector where each dimension set to 1 corresponds to the presence of one of the d possible labels observed in a history of length m, i.e. m steps before the target element x_i.

In order to apply a SVM, a projection function $\phi_m(\cdot)$ can be defined to consider both the observations x_i and the transitions $\psi_m(x_i)$ by concatenating the two representations as follows: $\phi_m(x_i) = x_i \mid\mid \psi_m(x_i)$ with $\phi_m(x_i) \in \mathbb{R}^{n+md}$. Notice that the symbol $\mid\mid$ here denotes the vector concatenation, so that $\psi_m(x_i)$ does not interfere with the original feature space, where x_i lies. Kernel-based methods can be applied in order to model meaningful representation spaces, encoding both the feature representing individual examples together with the information about the transitions. According to kernel-based learning [STC04], we can define a kernel function $K_m(x_i, z_j)$ between a generic item of a sequence x_i and another generic item z_j from the same or a different sequence, parametric in the history length m. It surrogates the dot product between $\phi_m(\cdot)$ such that:

$$K_m(x_i, z_j) = \phi_m(x_i)\phi_m(z_j) = K^{obs}(x_i, z_j) + K^{tr}\big(\psi_m(x_i), \psi_m(z_j)\big)$$

We define a kernel that is the linear combination of two further kernels: K^{obs} operating over the individual examples x_i and a K^{tr} operating over the feature vectors encoding the involved transitions. It is worth noticing that K^{obs} neither depends on the position nor on the context of individual examples, in line with Markov assumption that characterizes a large class of these generative models, e.g. HMM. For simplicity, we define K^{tr} as a linear kernel between input instances, i.e. a dot-product in the space generated by $\psi_m(\cdot)$:

$$K_m(x_i, z_j) = K^{obs}(x_i, x_j) + \psi_m(x_i)\psi_m(z_j)$$

At training time, we use the kernel-based SVM in a One-Vs-All schema over the feature space derived by $K_m(\cdot, \cdot)$. The learning process provides a family of classification functions $f(x_i; m) \subset \mathbb{R}^{n+md} \times \mathbb{R}^d$, which associate each x_i to a distribution of scores with respect to the different d labels, depending on the context size m. At classification time, all possible sequences $\mathbf{y} \in \mathcal{Y}^+$ should be considered in order to determine the best labeling $\hat{\mathbf{y}}$, where m is the size of the history used to enrich x_i, that is: $\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \in \mathcal{Y}^+} \{\sum_{i=1\ldots m} f(x_i; m)\}$

In order to reduce the computational cost, a *Viterbi-like decoding algorithm* is adopted[1] as described in Fig. 1. The next section defines the kernel function K^{obs} applied to specific turns in the drama.

2.2 Modeling dramatic properties as ML features

Three types of kernels are applied for different types of features.

Lexical features include *sentence embeddings* as linear combinations of individual word embeddings, *grammatical patterns*, such as verb-objects or subject-verb pairs, *POS n-grams* (n=3) and, finally, *sentence properties* such as *length* and *complexity* (e.g. number of different active mode verbs).

Narrative features are strictly dependent on the narrative structure and express possible CHARACTERS and ACTIONS in a turn. Named Entity Recognition is first run on the individual utterances, to capture character mentions. A narrative vector including the *acting character* (e.g. LOPAKHIN in line `0036` or `0038` in the Appendix) as well as all the other *recently mentioned characters* (e.g. LOPAKHIN and DUNYASHA in the editorial note at line `0042`). Individual features modeling the number of mentioned or recently mentioned characters for each turn will be adopted. An aging mechanism defines lower scores for no longer mentioned characters. Finally, narrative features denoting the ACTIONS mentioned in a turn will be adopted in order to account for the interaction (and possible conflicts) in an explicit way. Examples are motion verbs such as *to_come*, *to_go*, social verbs, such as *to_meet* (see LOPAKHIN in unit `0040`) or even emotional verbs (e.g. or *to_faint* as in unit `0041`.

[1] When applying $f(x_i; m)$ the classification scores are normalized through a softmax function and probability scores are derived.

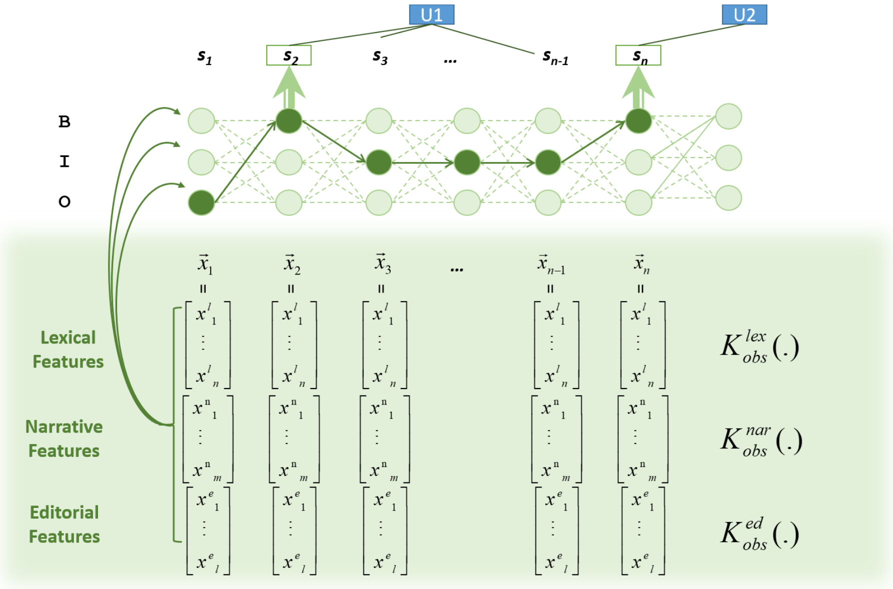

Figure 1: The overall sequence labeling architecture for event segmentation.

Specific dictionaries of English verbs and their nominalizations will be used here to denote narratively interesting ACTIONS.

Editorial features will depend on the material that includes the author's suggestions in the environment (see, for example, the sentence "*A room which is still called the nursery*" in the incipit). In this case, a representation that is similar to the one for the lexical features for individual acting turns is adopted, but the editorial material will be expressed through a separated vector, in order to play an independent role.

Table 1: Performance scores and ablation analysis for the segmentation based on different lexical features. Token-based Accuracy figures are *Strict* when applied only to B-labeled paragraphs in the oracle, and *Greedy* when also all the consistently aligned I-labeled paragraphs are considered as correct.

		Basic Editorial Features	**+ Simple Lexical Features**	**+ Lexical Contextual Features**	**+ Word Embedding Features**
B tag	Precision	13.9%	44.4%	50.0%	**52.1%**
	Recall	34.3%	35.8%	34.3%	**37.3%**
	F1	19.7%	39.7%	40.7%	**43.5%**
I tag	Precision	94.0%	94.9%	94.9%	**95.1%**
	Recall	82.9%	96.4%	97.3%	**97.3%**
	F1	88.1%	95.7%	96.0%	**96.2%**
Token-based Accuracy	Strict	17.3%	33.4%	36.5%	**40.5%**
	Greedy	33.0%	70.7%	75.8%	**77.0%**

3 Experimental Evaluation and Discussion

In the current experimental stage, we applied ablation analysis to the set of Lexical Features as described in the previous section. The lexical model is tested via the HMM_SVM framework (implemented within KeLP, [FCM+18]) on the annotated version of *The Cherry Orchard*, in its English translation: the Appendix reports a short excerpt. The work includes 4 acts made of about 904 paragraphs, segmented into 67 units. The different tokens in our labeled corpus $L = O_L$ are thus about 22,800. Every paragraph has been considered part of a sequence of length k=5 that corresponds to the local input to the tagger. Every paragraph in the sequence is represented via feature vectors and several lexical representations have been adopted:

- (*Simple Lexical*) A bag-of-word feature vector including lemmas, bi-grams and POS tags occurring in the target paragraph are represented.

- (*Baseline*) As a baseline, a set of simple heuristics from narrative features is used to simulate a blind typographic approach. The synthetic vector encodes only a label with the guessed editorial role of the paragraph. In this way, individual utterance and editorial notes are just kept separated.
- (*Contextualized lexical*) Similar to the *Lex* feature vector, but extended with the vector of the preceding paragraph, in order to contextualize the model.
- (*Word Embeddings*) A real-valued vector that corresponds to the sentence embedding of the target paragraph is adopted.

Each training paragraph is labeled according to the IOB notation, i.e. as `B` or `I`. A macro n-fold cross validation is applied with one fold per act, i.e $n = 4$. In one evaluation step, one act is removed from the dataset: training on the three remaining acts is carried out by leaving a 10% of paragraphs as development data (i.e. for tuning of the SVM parameters): the automatic tagging over the left-out act allows to measure and average the labeling accuracy. Each time, the training set is randomly split 90/10 to derive a development set used to tune the SVM parameters.

Measures of performance are class-based precision and recall, while accuracy is the percentage of paragraphs that are correctly re-labeled with respect to the original IOB label. Micro-average across the four different 4 folds is applied. Notice that for the unbalanced presence of the `I` tag (i.e. 92.6% of the paragraphs), the simple baseline model achieved a **93.5%** of accuracy across all paragraphs. For this reason, in Table 1, we just report precision, recall and F1 for the two separated classes. Moreover, we report the strict accuracy, just computing the accuracy restricted to the `B` gold labeled paragraph. Notice that this class is defined by only 67 positive examples in the training dataset. Finally, the accuracy measured on only the aligned `B` and `I` paragraph is reported: it considers correct a paragraph labeled as "inner" by the system only when this does not violate any boundary `B` in the oracle annotation. As Table 1 shows more complex lexical features brings more information as they increase performance for each measure. Moreover, (last column in last line) the token based accuracy suggests that the current model correctly annotate about 70% of the paragraphs of the work, thus representing a large advantage against manual annotation.

Examples of mistaken segmentations are hereafter reported where the gold and automatic labels are shown after the row number for the different paragraphs, respectively:

...
`801 I I` (*Goes off*)
`802 B I` *YASHA remains, sitting beside the shrine.*
`803 I B` *Enter RANYEVSKAYA, GAYEV, and LOPAKHIN.*
`804 I I` LOPAKHIN. It has to be settled once and for all - time won't wait. Look, it's a simple enough question. Do you agree to lease out the land for summer cottages or not? Answer me one word: yes or no? Just one word!
...
`901 I I` YEPIKHODOV (*Off, behind the door*). I'll tell about you!
`902 I I` VARYA. Oh, coming back, are you? (*Seizes the stick that FIRS left besides the door.*) Come on,then...Come on... Come on... I'll show you... Are you coming? My word, you're going to be for it...! (*Raises the stick threateningly.*)
`903 I B` *Enter LOPAKHIN.*
`904 B I` LOPAKHIN. Thank you kindly.
`905 I I` VARYA (*angrily and sarcastically*). Sorry! My mistake.
...

According to the gold labels, the `B`-labeled paragraphs in lines `803` and `903` are all wrong, while `801,804` and `901,902,905` are correctly aligned `I`-labeled paragraphs: these latter are retained in the greedy version of the Token-based Accuracy scores. Notice how the mistakes are mainly due to mismatches in the way editorial material is used by the human annotators. In the first example, lines `802, 803`, the beginning is annotated to the sitting act of YASHA (line `802`). In the second, the first speech of LOPAKHIN is used to start a new segment (line `904`). On the contrary, in both cases the system has focused on the entrance of the new character to suggest the start (i.e. `B` labels in line `803` and `903`).

These mild errors suggest how the generalization of the system at this current stage of development is already acceptable in several cases. Accuracy rates are thus expected to grow when more complex features (for example the narrative features that will better express the ontological information) will be adopted. This will be part of future work.

References

[BCV$^+$16] Emanuele Bastianelli, Danilo Croce, Andrea Vanzo, Roberto Basili, and Daniele Nardi. A discriminative approach to grounded spoken language understanding in interactive robotics. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2747–2753, 2016.

[CB11] Danilo Croce and Roberto Basili. Structured learning for semantic role labeling. In *AI*IA 2011: Artificial Intelligence Around Man and Beyond - XIIth International Conference of the Italian Association for Artificial Intelligence, Palermo, Italy, September 15-17, 2011. Proceedings*, pages 238–249, 2011.

[Che17] Anton Chekhov. *The Cherry Orchard.* Plays, by Anton Tchekoff. 2d series, tr. with an introduction by Julius West. New York, Scribner's, 1917.

[Cho00] F. Y. Choi. Advances in domain independent linear text segmentation. In *Proceedings of the 1st NAACL Conference*, pages 26–33. ACL, 2000.

[DLPar] Rossana Damiano, Vincenzo Lombardo, and Antonio Pizzo. The ontology of drama. *Applied Ontology*, to appear.

[FCM$^+$18] Simone Filice, Giuseppe Castellucci, Giovanni Da San Martino, Aless, ro Moschitti, Danilo Croce, and Roberto Basili. Kelp: a kernel-based learning platform. *Journal of Machine Learning Research*, 18(191):1–5, 2018.

[Hea94] Marti A. Hearst. Multi-paragraph segmentation of expository text. In *ACL*, pages 9–16. Morgan Kaufmann Publishers / ACL, 1994.

[KKM98] Min-Yen Kan, Judith L. Klavans, and Kathleen R. McKeown. Linear segmentation and segment significance. In *VLC@COLING/ACL*, 1998.

[LDP18] Vincenzo Lombardo, Rossana Damiano, and Antonio Pizzo. Drammar: A comprehensive ontological resource on drama. In *ISWC 2018 - 17th Int. Semantic Web Conf., Monterey, CA, USA, October 8-12, 2018, Proceedings, Part II*, pages 103–118, 2018.

[LPD16] Vincenzo Lombardo, Antonio Pizzo, and Rossana Damiano. Safeguarding and accessing drama as intangible cultural heritage. *JOCCH*, 9(1):5:1–5:26, 2016.

[LSJ18] Jing Li, Aixin Sun, and Shafiq Joty. Segbot: A generic neural text segmentation model with pointer network. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4166–4172. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781, 2013.

[Rey94] Jeffrey C. Reynar. An automatic method of finding topic boundaries. In *ACL*, pages 331–333. Morgan Kaufmann Publishers / ACL, 1994.

[SDDK11] Fei Song, William M. Darling, Adnan Duric, and Fred W. Kroon. An iterative approach to text segmentation. In *ECIR*, volume 6611 of *Lecture Notes in Computer Science*, pages 629–640. Springer, 2011.

[STC04] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[SZR07] N. K. Speer, J. M. Zacks, and J. R. Reynolds. Human brain activity time-locked to narrative event boundaries. *Psychological Science*, 18(5):449–455, 2007.

[TJHA05] Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. Large margin methods for structured and interdependent output variables. *J. Machine Learning Reserach.*, 6, 2005.

[Yaa99] Yaari Yaakov. Segmentation of expository text by hierarchical agglomerative clustering. In *Recent Advances in NLP (RANLP'97)*. ACL, 1999.

[YXX+16] J. Yu, X. Xiao, L. Xie, E. S. Chng, and H. Li. A dnn-hmm approach to story segmentation. *INTERSPEECH*, pages 1527–1531, 2016.

[ZS07] J. M. Zacks and K. M. Swallow. Event segmentation. *Current Directions in Psychological Science*, 16(2):80–84, 2007.

[ZSSM10] J. M. Zacks, N. K. Speer, K. M. Swallow, and C. J. Maley. The brain's cutting-room floor: Segmentation of narrative cinema. *Frontiers in human neuroscience*, 4, 2010.

[ZT01] J. M. Zacks and B. Tversky. Event structure in perception and conception. *Psychological bulletin*, 127(1):3, 2001.

Appendix: a segmentation example: unit 4-6

From *The Cherry Orchard* by Anton Checkov

0034 YEPIKHODOV. I'll go, then. *(Stumbles against the table, which falls over.)* There you are... *(As if exulting in it.)* You see what I'm up against! I mean, it's simply amazing! *(Goes out.)*

`UNIT ID: 0004, UNIT NAME: Dunyasha struts around, SPAN:35,37`

0035 DUNYASHA. To tell you the truth, he's proposed to me.

0036 LOPAKHIN. Ah!

0037 DUNYASHA. I don't know what to say... He's all right, he doesn't give any trouble, it's just sometimes when he starts to talk you can't understand a word of it. It's very nice, and he puts a lot of feeling into it, only you can't understand it. I quite like him in a way, even. He's madly in love with me. He's the kind of person who never has any luck. Every day something happens. They tease him in our part of the house - they call him Disasters by the Dozen...

`UNIT ID: 0005, UNIT NAME: Lopakhin and Dunyasha welcome the masters, SPAN:38,42`

0038 LOPAKHIN(*(listens).* I think they're coming.

0039 DUNYASHA. They're coming! What's the matter with me? I've gone all cold.

0040 LOPAKHIN. They are indeed coming. Let's go and meet them. Will she recognize me? Five years we haven't seen each other.

0041 DUNYASHA *(in agitation).* I'll faint this very minute... I will, I'll faint clean away!

0042 *Two carriages can be heard coming up to the house. LOPAKHIN and DUNYASHA harry out. The stage is empty.*

`UNIT ID: 0006, UNIT NAME: The owners settle down, SPAN: 43, ...`

...