NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

PROGRAM OF POSTGRADUATE STUDIES

MASTER'S THESIS

Extending the temporal tagger HeidelTime for the Greek language

Emmanouil I. Kapernaros

SUPERVISOR **Manolis Koubarakis,** Professor

ATHENS

JULY 2020

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επεκτείνοντας τον χρονικό σχολιαστή HeidelTime για την Ελληνική γλώσσα

Εμμανουήλ Ι. Καπερνάρος

Επιβλέπων **Μανόλης Κουμπαράκης,** Καθηγητής

ΑΘΗΝΑ

Ιούλιος 2020

MASTER'S THESIS

Extending the temporal tagger HeidelTime for the Greek language

Emmanouil I. Kapernaros
A.M.: PIB136

SUPERVISOR: **Manolis Koubarakis,** Professor

EXAMINING COMMITTEE: **Gunopulos Dimitrios,** Professor

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επεκτείνοντας τον χρονικό σχολιαστή HeidelTime για την Ελληνική γλώσσα

Εμμανουήλ Ι. Καπερνάρος
Α.Μ.: ΠΙΒ 136

ΕΠΙΒΛΕΠΩΝ: **Μανόλης Κουμπαράκης,** Καθηγητής

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ: **Γουνόπουλος Δημήτριος,** Καθηγητής

ABSTRACT

Here we describe our work on extending the temporal tagger *HeidelTime* for the Greek language. *HeidelTime* is a Multilingual rule-based Temporal Tagger that performs the full task of temporal tagging including the extraction and normalization subtasks. It achieves multilingual capabilities by using language-specific resources that are separate from its source code and can be easily adapted. It contains manually developed resources for 13 languages and automatic resources for 200+ languages including Greek. *HeidelTime* can extract and then classify temporal expressions into *date*, *time*, *duration* and *set* classes and then normalize them with a standard format value. For instance, when the expression "March 11, 2013" is detected, it is extracted and normalized with the value “2013-03-11”. The goal of the project is to manually develop publicly available Greek Resources extending the automatic ones. We did this developing language specific resources which are *.txt* files with a specific syntax. For the development process a manually annotated Greek corpus was necessary for the training. For this purpose we created *WikiWarsEL,* a corpus with Greek annotated expressions which contains 19 war documents from Greek *Wikipedia*. Finally, we evaluated the newly developed resources with more war documents from *Wikipedia* which were not used in the training process. The Value F1-score result we achieved was 82.31%, a significant improvement over the 2.19% of the automatic resources.

SUBJECT AREA: Natural language processing

KEYWORDS: time, temporal expressions, temporal tagger

ΠΕΡΙΛΗΨΗ

Εδώ περιγράφουμε την δουλειά μας για την επέκταση του λογισμικού σχολιασμού χρονικών εκφράσεων HeidelTime για υποστήριξη της Ελληνικής γλώσσας. Το HeidelTime είναι ένας πολυγλωσσικός χρονικός σχολιαστής που λειτουργεί βάση κανόνων ο οποίος εκτελεί την πλήρη διαδικασία σχολιασμού συμπεριλαμβανομένων της εξαγωγής και κανονικοποίησης. Επιτυγχάνει πολυγλωσσικότητα με ειδικούς γλωσσικούς πόρους οι οποίοι είναι διαχωρισμένοι από τον πηγαίο κώδικά του και μπορούν εύκολα να προσαρμοστούν. Περιλαμβάνει χειροκίνητα ανεπτυγμένους πόρους για 13 γλώσσες και αυτόματα ανεπτυγμένους για περισσότερες από 200 γλώσσες συμπεριλαμβανομένης και της Ελληνικής. Το HeidelTime μπορεί να εξάγει χρονικές εκφράσεις και να τις ταξινομεί σε ημερομηνία, ώρα, διάρκεια και σύνολα και μετά να τις κανονικοποιεί με μια τυποποιημένης μορφής τιμή. Για παράδειγμα, όταν ανιχνευθεί η έκφραση “13 Μαρτίου 2013”, εξάγεται και κανονικοποιείται με την τιμή “2013-03-11”. Ο σκοπός αυτή της εργασίας είναι να αναπτύξουμε δημόσια διαθέσιμους χειροκίνητους Ελληνικούς πόρους επεκτείνοντας τους αυτόματα ανεπτυγμένους. Αυτό το κάναμε αναπτύσσοντας ειδικούς γλωσσικούς πόρους οι οποίοι είναι *.txt* αρχεία με μια συγκεκριμένη σύνταξη. Για την διαδικασία της ανάπτυξης ήταν απαραίτητο ένα Ελληνικό σώμα για την εκπαίδευση. Για τον σκοπό αυτό δημιουργήσαμε το WikiWarsEL, ένα σώμα με Ελληνικές σχολιασμένες εκφράσεις που περιέχει 19 πολεμικά κείμενα από την Ελληνική *Wikipedia*. Τέλος, αξιολογήσαμε τους νέους πόρους με περισσότερα πολεμικά κείμενα από τη Wikipedia τα οποία δεν χρησιμοποιήθηκαν κατά την εκπαίδευση. Το αποτέλεσμα του στατιστικού μέτρου F1-score ήταν 82.31%, μια σημαντική βελτίωση από το 2.19% των αυτόματων πόρων.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Επεξεργασία Φυσικής Γλώσσας

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: χρόνος, χρονικές εκφράσεις, χρονικός σχολιαστής

ΣΥΝΟΠΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Η εξαγωγή (extraction) και κανονικοποίηση (normalization) χρονικών εκφράσεων (temporal expressions) είναι σημαντικά μέρη για πολλά συστήματα αναγνώρισης φυσικής γλώσσας. Συστήματα επεξεργασίας φυσικής γλώσσας χρησιμοποιούνται για διάφορες εφαρμογές όπως εξαγωγή πληροφοριών, χρονολογική ταξινόμηση γεγονότων και απάντηση ερωτήσεων. Χρονικές πληροφορίες μπορούμε να βρούμε στα μεταδεδομένα (metadata) εγγράφων όπως η ημερομηνία δημιουργίας ή τροποποίησης αυτών των εγγράφων αλλά και στο περιεχόμενο κείμενο όπως σύνθετες εκφράσεις για τον καθορισμό γεγονότων.

Ο χρονικός σχολιασμός (temporal tagging) είναι η διαδικασία που περιλαμβάνει την εξαγωγή και την κανονικοποίηση των χρονικών εκφράσεων σε κάποιο κείμενο. Αυτή η διαδικασία μπορεί να υλοποιηθεί με μεθόδους που κάνουν χρήση νευρωνικά δίκτυα, αλλά μπορεί να υλοποιηθεί και βάση κανόνων (rule- based) οι οποίοι ορίζουν λεξικά και συντακτικά κριτήρια για την αναγνώριση των χρονικών εκφράσεων. Σε αυτή την εργασία θα ασχοληθούμε με τη δεύτερη μέθοδο και συγκεκριμένα με το λογισμικό HeidelTime [1] το οποίο αναπτύχθηκε στο Πανεπιστήμιο της Χαϊδελβέργης και χρησιμοποιεί κανόνες που αποτελούνται από κανονικές εκφράσεις (regular expressions) για την εξαγωγή και κανονικοποίηση. Το HeidelTime είναι ένας πολυγλωσσικός (multilingual) χρονικός σχολιαστής (temporal tagger) διάφορων τομέων (cross-domain). Επιτυγχάνει πολυγλωσσικότητα με ειδικούς για κάθε γλώσσα πόρους (resources) οι οποίοι είναι διαχωρισμένοι από τον πηγαίο κώδικά του. Στους πόρους αυτούς περιλαμβάνονται οι κανόνες (rules) που αναφέραμε προηγουμένως, τα πρότυπα (patterns) εκφράσεων για την εξαγωγή και οι τιμές (values) κανονικοποίησης. Οι τομείς τους οποίους υποστηρίζει το HeidelTime είναι η αφήγηση (narrative) για έγγραφα όπως είναι τα ιστορικά και τα νομικά, η ειδησιογραφία (news), η καθομιλουμένη (colloquial) και η επιστήμη (scientific). Στη παρούσα εργασία θα ασχοληθούμε με κείμενα αφήγησης.

Οι δυο ορισμοί στους οποίους θα αναφερόμαστε συχνά καθ' όλη την έκταση της εργασίας αυτής είναι η **χρονική έκφραση** και η **τιμή** (value). Χρονική έκφραση είναι είτε μια έκφραση που αναφέρεται σε μία ημερομηνία (π.χ. τον Μάιο του 2020) οποιασδήποτε λεπτομερειακότητας (granularity), μια έκφραση που αναφέρεται σε μια διάρκεια (π.χ. 5 ώρες) ή μια έκφραση που αναφέρεται σε μια περιοδικότητα (π.χ. κάθε μέρα). Αυτές οι εκφράσεις κατατάσσονται σε:

- **ρητές** (explicit) όταν περιλαμβάνουν ολόκληρη την πληροφορία για την τιμή της κανονικοποίησης χωρίς να εξαρτώνται από άλλο μέρος του περιεχόμενου κειμένου και χωρίς να χρειάζονται καμία επιπλέον γνώση
- **έμμεσες** (implicit) όταν η τιμή τους καθορίζεται από γνώση εκτός του περιεχομένου όπως οι γιορτές (π.χ. τα Χριστούγεννα εννοείται ότι είναι στις 25 Δεκεμβρίου)
- **σχετικές** (relative) όταν η τιμή τους καθορίζεται από τα συμφραζόμενα και πρέπει να αναζητηθεί ένα χρονικό σημείο αναφοράς στο περιεχόμενο κείμενο ενώ η σχέση μεταξύ της έκφρασης και του σημείου αναφοράς συνεπάγεται από το κείμενο (π.χ η φράση "το προηγούμενο έτος" κανονικοποιείται σε 1999 εάν προηγουμένως έχει αναφερθεί το έτος 2000)
- **υπο-ορισμένες** (underspecified) όταν ούτε η σχέση μεταξύ σημείου αναφοράς ούτε το σημείο αναφοράς είναι γνωστά, τότε η κανονικοποιημένη τιμή αποτελείται από "X" (π.χ. για την έκφραση "25 Δεκεμβρίου" η τιμή είναι XXXX-12-25)

Το HeidelTime περιλαμβάνει πόρους για περισσότερες από 200 γλώσσες οι οποίοι έχουν παραχθεί με αυτοματοποιημένο τρόπο και δεν παρέχουν την ίδια ποιότητα με τους χειροποίητους οι οποίοι αποτελούν ένα σύνολο από 13 γλώσσες τη στιγμή που

γράφεται αυτό το κείμενο. Ο σκοπός αυτής της διπλωματικής εργασίας είναι να επεκτείνουμε το HeidelTime αναπτύσσοντας πόρους για την Ελληνική γλώσσα και να τους διαθέσουμε δημόσια για συμπερίληψη τους στο HeidelTime.

Για το έργο αυτό αναπτύξαμε ένα σώμα (corpus) με Ελληνικά κείμενα από την Βικιπαίδεια στα οποία εφαρμόσαμε χειροκίνητο σχολιασμό των χρονικών εκφράσεων για να το χρησιμοποιήσουμε ως σύνολο εκπαίδευσης (training set) κατά την ανάπτυξη των Ελληνικών πόρων. Το σώμα αυτό το ονομάσαμε *WikiWarsEL* μιας και αποτελείται από έγγραφα με ιστορικές περιγραφές διάφορων πολέμων. Η δουλειά για το *WikiWarsEL* καλύπτει το ήμισυ αυτής της εργασίας. Επειδή δεν υπάρχουν επίσημες οδηγίες σχολιασμού για την Ελληνική γλώσσα η οποία παρουσιάζει μορφολογική ποικιλία, αναπτύχθηκαν μερικές συμβάσεις για τον τρόπο σχολιασμού της Ελληνικής γλώσσας. Ο σχολιασμός των χρονικών εκφράσεων του σώματος πραγματοποιήθηκε από ένα άτομο το οποίο διάβασε τα έγγραφα τουλάχιστον δυο φορές για να ελαχιστοποιηθεί η πιθανότητα παράληψης κάποιων εκφράσεων. Το αποτέλεσμα είναι ένα σώμα περίπου 77 χιλιάδων λέξεων με χειροκίνητα σχολιασμένες 1575 χρονικές εκφράσεις.

Το δεύτερο μισό της εργασίας αποτέλεσε την ανάπτυξη των Ελληνικών πόρων για το HeidelTime μέσα από μια επαναλαμβανόμενη διαδικασία δοκιμής και βελτίωσής τους προσπαθώντας να ταιριάξουμε όλο και περισσότερες εκφράσεις του συνόλου εκπαίδευσης. Ξεκινήσαμε βελτιώνοντας τους πόρους που παρέχει το HeidelTime για την Ελληνική γλώσσα οι οποίοι έχουν παραχθεί με αυτοματοποιημένο τρόπο, διορθώνοντάς τους και επεκτείνοντας τους. Αναπτύξαμε κανονικές εκφράσεις για την εξαγωγή και κανονικοποίηση λέξεων και εκφράσεων που εμφανίζονται στο σύνολο εκπαίδευσης και στη συνέχεια κανόνων που ορίζουν πότε πρέπει να γίνεται σχολιασμός χρονικής έκφρασης. Για παράδειγμα, με τις κανονικές εκφράσεις εξαγωγής και κανονικοποίησης μπορούμε να ορίσουμε ότι η λέξη “Ιανουάριος” κανονικοποιείται στην τιμή 01 και με τη χρήση κανόνων που ελέγχουν το γειτονικό κείμενο για περισσότερες λέξεις που ενδεχομένως να ανήκουν στην ίδια έκφραση, τελικά παράγεται η τιμή στην τυποποιημένη (standard) μορφή σύμφωνα με το ISO 8601.

Το αποτέλεσμα ήταν η ανάπτυξη σημαντικά ποιοτικότερων πόρων από τους αυτόματα παραγμένους με αύξηση της τιμής στατιστικού μέτρου *F1 εξαγωγής & κανονικοποίησης* από **2.19%** σε **82.31%** κατά την αξιολόγηση (evaluation) σε έγγραφα που δεν είχαν χρησιμοποιηθεί κατά την εκπαίδευση. Τα έγγραφα αυτά που χρησιμοποιήθηκαν για την αξιολόγηση αποτελούνται από κείμενα σχετικά με πολέμους επίσης από τη Βικιπαίδεια, συνολικού μεγέθους 18 χιλιάδων λέξεων και 440 χειροκίνητα σχολιασμένων χρονικών εκφράσεων ακολουθώντας την ίδια διαδικασία και τις συμβάσεις σχολιασμού που ακολουθήθηκαν για το σώμα *WikiWarsEL*.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

The present thesis is part of the requirements for the acquisition of a Master's degree in the Department of Informatics and Telecommunications of the National and Kapodistrian University of Athens (UoA), in cooperation with the Technological Educational Institute (TEI) of Athens, and in collaboration with the Foundation for Biomedical Research of the Academy of Athens (BRFAA) and the Institute of Informatics and Telecommunications of the National Centre for Scientific Research "Demokritos".

The goal is to develop an automatic process to detect temporal expressions in Greek text documents. For example, expressions like "one hundred years" should be extracted and then normalized with a standard format like "P100Y" which stands for Period of 100 Years in ISO 8601. Then temporal information can be easily passed to software programs and used in various ways.

This task involved working with the Greek languages' structure which was an experience that made me realize the complexity of natural languages and also gave me an insight of how natural language processing and Artificial Intelligence in general is an important part of knowledge.

1. INTRODUCTION

Extraction and Normalization of temporal expressions is an important part of Natural Language Processing. It is the task of detecting the expressions which carry a temporal meaning and then understanding and conveying this meaning in a computer readable format. Expressions of time and date are what mainly constitute temporal expressions in formal or colloquial parts of written text. Other types of temporal expressions include duration and frequency of events. All these different types of expressions are easily understandable by humans but it is a difficult challenge for computers. A word like “today” is simple for us to understand and deduce the exact date but a complete different story for computers.

There are various different approaches that accomplish the task of temporal expression tagging. One approach is using Machine Learning with Neural Networks to train a system on recognizing temporal expressions. This approach can be designed in various ways with different Neural Network arrangements. Another approach to this problem is a rule-based system that follows explicit manually developed rules. It is also possible to combine Machine Learning and rule-based approaches and use one for extraction and the other for normalization.

In this work we use *HeidelTime*[4] which is a solely rule-based Temporal Tagger that performs the full task of temporal tagging including the extraction and normalization sub-tasks. It achieves multilingual capabilities by using language-specific resources that are separate from its source code and can be easily adapted. It contains manually developed resources for 13 languages and automatic resources for 200+ languages including Greek. The resources are files that define the rules which must be followed by the system to extract and normalize temporal expressions for a specific language.

HeidelTime can extract and classify temporal expressions into *date*, *time*, *duration* and *set* classes and then normalize them with a standard format value. In particular for dates, it can work with explicit expressions like "March 11, 2013", implicit "Christmas 2013", relative "the following year" and under-specified "December". All these are dictated by the language specific resources which are text files easily editable and adaptable to different languages.

The goal of the project is to manually develop publicly available Greek resources extending the automatic ones. To develop the Greek resources we needed a manually annotated corpus of Greek text so that we can use it for training. This training process involved the development of the resources necessary to successfully extract and normalize as many temporal expressions as possible in the corpus. We developed resources to match certain expressions or groups of similar expressions that occurred in our training set.

The corpus that we used for the training process is a Greek annotated corpus with 19 war documents from *Wikipedia.* The Greek corpus we developed is *WikiWarsEL* and was inspired by *WikiWars*, *WikiWarsDE* and *WikiWarsVN* which are the English, German and Vietnamese versions of the corpus. The development of this corpus was started out of the necessity to have a frame of reference for temporal expressions. So our work consists mainly of two parts. The first part is the development of the Greek corpus and the second part is the development of the Greek resources for *HeidelTime*. The development of the corpus was done by one Annotator who read the untagged texts and tagged it following the guidelines of *TimeML* (Time Markup Language).

The complete text of the corpus consists of about 76.868 words and the Annotator found and tagged 1.575 temporal expressions. All text was read two times and some of them more than two. This was done because we observed the Annotator’s blindness

phenomenon where some of the temporal expressions were missed by the Annotator. Also, after completing the annotation, the automatic tagging of the system revealed errors in the Annotator's manual work which we fixed. After some iterations the corpus reached a stable period where no modifications were necessary to continue our work. This corpus is also made publicly available.

The organization of the rest of the dissertation is as follows. Section 2. presents background knowledge on which our work is dependent such as the basics of temporal annotation, the structure of *HeidelTime*, the software versions and the tools we used. Section 3. presents the development process of the Greek manually annotated corpus *WikiWarsEL* and Section 4. presents the development processes of the Greek language resources for *HeidelTime*. Finally, Section 5. concludes the dissertation.

2. BACKGROUND

2.1 The Annotation Task

The interest in temporal tagging has resulted in the creation of standards that can be used for the manual or automatic annotation. Two of the standards are the *TIDES TIMEX2* [1] and the *TimeML* [2] (Time Markup Language). They define what and how exactly to annotate text with temporal meaning. The extent of an expression as well as the standardized format value that must be assigned to it are parts of these standards.

TimeML extends *TIMEX2* with many changes and additions which make the two standards incompatible to each other and therefore it uses the *TIMEX3* tag. Our work in this project did not involve the annotation with tags other than the *TIMEX3* tag. We focused only in annotating temporal expressions so we skipped most of the guidelines about the *EVENT*, *SIGNAL*, *MAKEINSTANCE*, *TLINK*, *ALINK*, and *SLINK* tags which are also defined in *TimeML*.

When an expression in natural language text is to be annotated like the expression “5 days” in the text “it was raining for 5 days”, we wrap it around the <*TIMEX3*> tag. There are three mandatory attributes for this tag. The attribute *tid* which is an integer which uniquely identifies the expression, the attribute *type* which can be of either *DATE*, *TIME*, *DURATION* or *SET* value accordingly and the *value* attribute which holds the normalized value. In our example the annotated text becomes:

> It was raining for <TIMEX3 tid=”t1” type=”DURATION” value=”P5D”>5 days</TIMEX3>.

The value “P5D” (which stands for Period of 5 Days) and the *value* attribute in general follows the standard format ISO 8601. There are more attributes defined for *TIMEX3* for other uses some of which will be described in Section 3..

2.2 HeidelTime Temporal Tagger

HeidelTime is a Temporal Tagger developed at the University of Heidelberg. It is publicly available and offers natural language processing of many languages provided that there are language specific resources already developed for it. It is designed to make the development of these resources easy by decoupling them from its source code. That means adding support for a language can be accomplished by developing a number of *.txt* files related to the grammar and syntactic of this specific language.

The job of HeidelTime is the Extraction and Normalization of temporal expressions in natural language. These two are the sub-tasks that constitute the full task of Temporal tagging. Temporal Tagging can be done using either a neural network approach or a rule-based approach. HeidelTime is a rule-based Temporal Tagger using rules contained in the resources mentioned above. In the extraction phase it detects expressions with temporal meaning and in normalization phase it assigns the temporal meaning in the ISO 8601 standard format.

2.2.1 Types of Temporal Expressions

Temporal expressions can be explicit, implicit, relative or under-specified. HeidelTime handles these cases differently according to the domain of the document(s). The domain of the document(s) can be news, narratives, scientific or colloquial and in any case this is set accordingly in configuration before runtime. First let’s discuss the different expression types.

An **explicit expression** contains all temporal information needed for normalization. For example the expression “March 14, 1998 “can be normalized to “1998-03-14” without the need of any other context.

An **implicit expression** does not provide all the information for normalization because it relies on the general knowledge of the reader. An example is “Christmas” which implies December 25 without mentioning this date anywhere in the text.

A **relative expression** can be normalized only in relation to another expression. This is the main challenge for HeidelTime as it has to find the correct reference of a relative expression to deduce the missing temporal meaning. For example “the previous year” can be normalized to “1999” if the year 2000 is mentioned earlier in the text which can be used as reference.

Under-specified expressions are those expressions that the full temporal meaning can not be deduced from the context because there is not any reference expressions nor relations to expressions. For example this is true for the expression “Monday” where it is not clear which year, month or day is referring to.

2.2.2 Normalization of Relative Expressions

When an expression cannot be fully normalized without context, HeidelTime will search for and use this context according to the domain in which it is set to operate. Two of the domains that are supported are narrative documents and news documents. Figure 1 is used exactly as published in [4] to get an intuitive idea of how HeidelTime normalizes relative expressions.

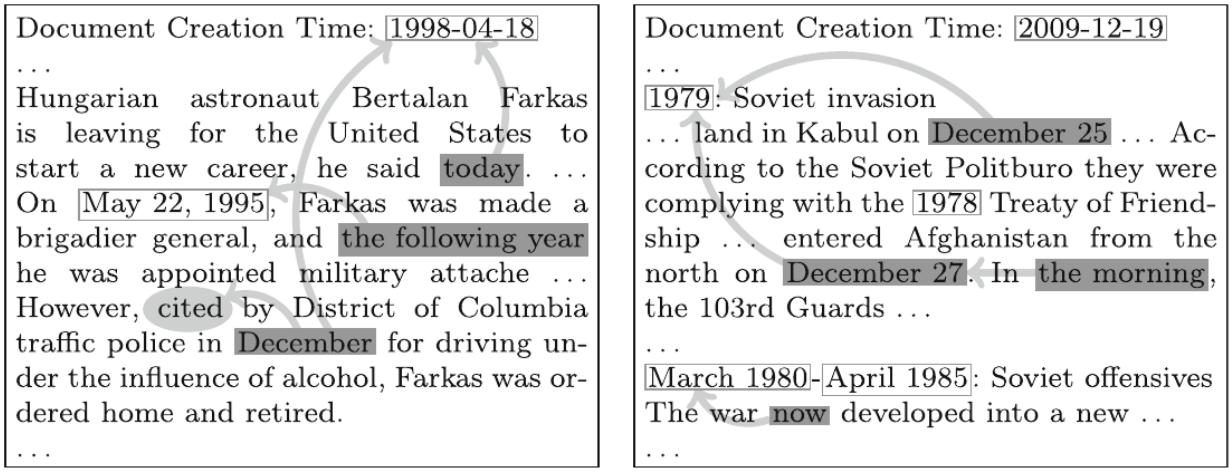

Figure 1: HeidelTime Relative Expressions Normalization, (a) for news, (b) for narratives

The expressions highlighted in dark boxes are relative expressions and context is necessary to normalize them. Expressions in white boxes are explicit expressions. In the first scenario (a) where the document’s domain is news, the tense of the verb “cited” is used to determine that “December” refers to December 1997 instead of December 1998. To normalize the expression “the following year”, HeidelTime uses the previous explicit expression “May 22, 1995” and to normalize “today” it uses the document’s creation time.

For narratives (b), this time, HeidelTime looks backwards for a reference expression which can only be an explicit expression.

2.2.3 Language Resources

Language specific resources are *.txt* files divided in three subdirectories, *pattern*, *normalization* and *rules* according to its purpose. *Pattern* files contain one regular expression in each line for use in the extraction phase. *Normalization* files contain one regular expression in each line for use in the normalization phase and *rules* files contain a rule in each line of how the extraction and normalization phases should be performed.

Pattern files

These files are used to create groups of words within a same meaning category and/or different forms of a word instance to make the extraction phase simpler and more efficient. For instance, the word “ημέρα” (day) can take different forms according to the number and the case as demonstrated in Table 1.

Table 1: Forms of the word "day" in different Numbers and Cases

	Singular	**Plural**
Nominative	ημέρα	ημέρες
Genitive	ημέρας	ημερών
Accusative	ημέρα	ημέρες

Also, this word can be written without the first letter “η”. Consequently, the following three lines with regular expressions where included in a file called *resources_repattern_reUnit.txt*:

η?μέρας?
η?μέρες
η?μερών

This method gives us the ability to use an abstract concept of a group instead of complicated regular expressions in the extraction part when developing the rules.

Normalization files

To assign a value on the extracted expression we use the *normalization* files. Inside these files each line contains a regular expression pattern and a value. We can have multiple files with the same regular expression patterns but a different value. For example, the word “day” can be used to denote a duration but also a particular point in time, depending on the context.

An example is shown in Table 2 where two *normalization* files are used for the word “ημέρα” (day), to normalize it with the value “D” for duration and “day” when a particular calendar day has to be determined by HeidelTime. When developing a rule for duration expressions we can use *resources_normalization_normUnit4Duration.txt* while for points in time we can use *resources_normalization_normUnit.txt*.

Table 2: Example of Normalization files

Filename:	**resources_normalization_normUnit4Duration.txt**	**resources_normalization_normUnit.txt**

Pattern and Value:	"η?μέρας?","D" "η?μέρες","D" "ημ?ερών","D"	"η?μέρας?","day" "η?μέρες","day" "η?μερών","day"

Rules

Finally, the missing piece in the resources puzzle are the *rules* files. Each line in the *rules* files puts together the extraction and normalization phases for a particular type of expression. Continuing the previous examples, let's say we want to build a rule for the type of expressions like "5 ημέρες" (5 days). Following the correct syntax we write a regular expression in the *EXTRACTION* field which matches an integer of one or more digits, a space and the patterns inside the *pattern* file *resources_repattern_reUnit.txt* from the previous example. Then we capture the values of the extracted regular expression groups and compose a value in standardized format as seen in the *NORM_VALUE* field below:

```
RULENAME="duration_example_rule",
EXTRACTION="([\d]+) %reUnit",
NORM_VALUE="Pgroup(1)%normUnit4Duration(group(2))"
```

This rule will return the normalized value "*P5D*" of the example expression. At this point we can expand *pattern* and *normalization* files with the inclusion of more unit-type words like week, month and year, so that the rule can become more general.

2.3 Installation

2.3.1 Software Versions

We installed *heideltime-kit-2.2.1* from the official github repository[1] following the instructions in the *README* file. We used *Debian 10* for the operating system with *OpenJDK 1.8.0_222* installed from the *oldstale* repository as the default Java Runtime Environment and *Python 2.7.16* as the default Python Interpreter.

Because some of the links that were provided in the *README* file were no longer valid, we used the most recent versions at the time of writing as an alternative to UIMA[2], TreeTagger[3], TreeTagger scripts[4], TreeTagger english parameters[5], TreeTagger german parameters[6] and TreeTagger Greek parameters[7] files.

2.3.2 Reproducing German evaluation results

To get familiar with the system and test its operation we followed the instructions in the wiki[8] to reproduce the evaluation results on the German *WikiWarsDE* corpus.

---

[1] https://github.com/HeidelTime/heideltime

[2] http://archive.apache.org/dist/uima/uimaj-2.6.0/uimaj-2.6.0-bin.tar.gz

[3] https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger-linux-3.2.2.tar.gz

[4] https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tagger-scripts.tar.gz

[5] https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/english.par.gz

[6] https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/german.par.gz

[7] https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/greek.par.gz

[8] https://github.com/HeidelTime/heideltime/wiki/Reproducing-Evaluation-Results

The results from the generated file *evaluation_results/wikiwarsde/evaluation_results.txt* are shown in Table 3 and are about the same as the published results on the wiki. The maximum difference we observe is 0.4% which is probably due to the difference in the software versions we used.

An important metric is the F-score of Extraction & Normalization (lenient) which shows us how how well the system performs while taking into account both the ability to correctly match expressions and assign them the correct values. As we can see this metric is 83.3 for German.

Table 3: German Evaluation Results

	Precision	**Recall**	**F-score**
Extraction (lenient)	98.7%	89.7%	94.0%
Extraction (strict)	92.6%	84.1%	88.2%
Normalization (value)	88.6%	88.6%	88.6%
Extraction \& Normalization (lenient + VAL)	87.5%	79.5%	83.3%
Extraction \& Normalization (strict + VAL)	83.3%	75.7%	79.3%

2.3.3 Development Environment for Greek Resources

HeidelTime contains automatically created resources for 200+ languages including Greek. We started developing our manually created resources building upon *auto-greek* resources, by improving and extending them.

We began by creating a similar directory structure as in the evaluation process to take advantage of the already existing scripts and tools for testing the performance of Greek resources during the development process.

We started with the latest automatic Greek resources (as of Oct. 2017) and we modified the scripts to support Greek characters and be more verbose for debugging purposes. In particular we replaced *ascii* with *utf8* inside the script *evaluate_entities.py*, added support for the *wikiwarsel* option to *evaluate_corpus_tempeval3style.sh* and created *wikiwarsel_workflow.xml* using the following settings:

* Collection Reader: TempEval-3 Reader
* Input Directory: ../corpora/WikiWarsEL/untagged
* Annotate Creation Time: true
* Analysis Engine: TreeTaggerWrapper
* Language: greek
* Annotate_tokens: true
* Annotate_partofspeech: true
* Annotate_sentences: true
* Chinese Tokenizer Path: empty
* Analysis Engine: HeidelTime
* Language: greek
* Date: true
* Time: truewith cpeGui.sh
* Duration: true
* Set: true

* Type: narratives
* CAS Consumer: TempEval-3 Writer
* Output Directory: ../uima_output/wikiwarsel

2.4 Annotation Tools

We started the manual annotation of *WikiWarsEL* (discussed in Section 3.) using the latest version *Callisto*[9] software. We completed 3 out of 19 documents using this software but due to some bugs and the fact that *Callisto* was no longer supported we developed custom *VIM* scripts[10] for the annotation of rest of the documents. A preview of the annotation task using our custom scripts can be seen in Figure 2.

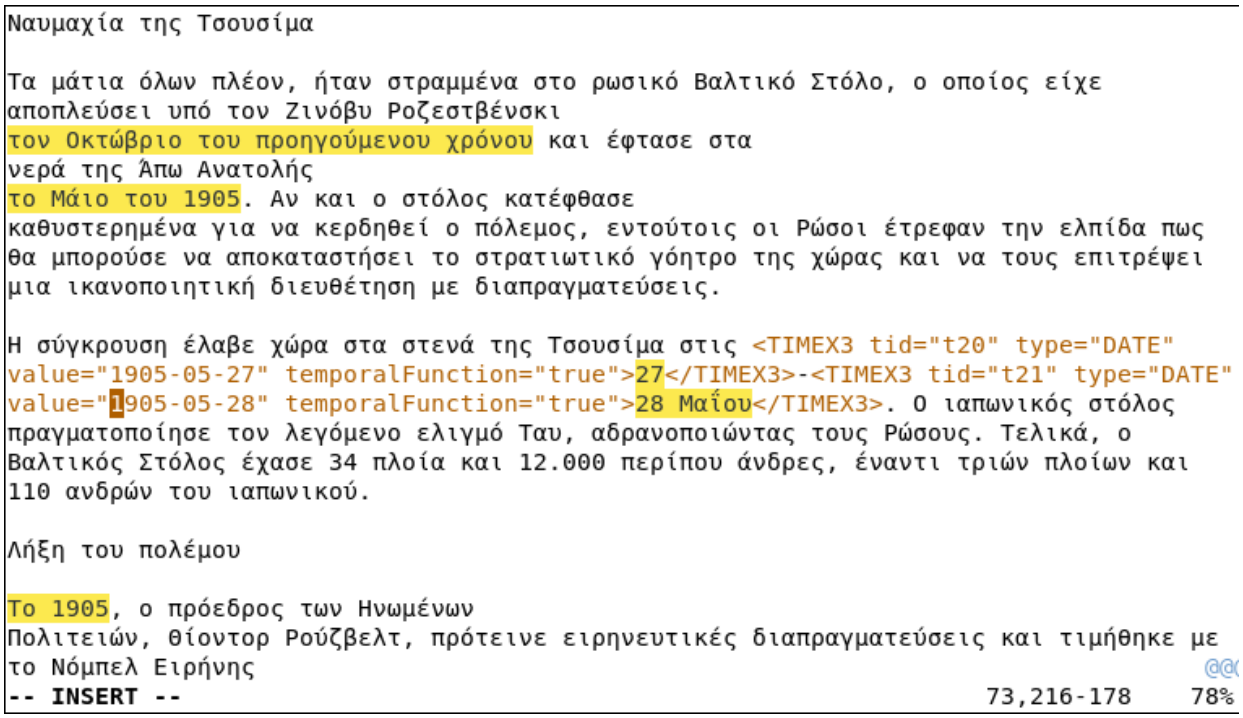

Figure 2: Example screen using VIM for annotation

[9] https://mitre.github.io/callisto/

[10] https://github.com/mkapernaros/vim-timex-annotation/

3. CORPUS DEVELOPMENT

The work on *WikiWarsEL* was based on the Vietnamese *WikiWarsVN* [5], a narrative-style corpus with <*TIMEX3*> annotated temporal expressions. The Vietnamese corpus is based on the English *WikiWars*[7] corpus which is a corpus of 22 most famous wars in history but annotated with <*TIMEX2*> tag. We used *WikiWarsVN* as a template for our Greek version. We downloaded the same war articles from the Greek *Wikipedia*[11] except the ones about the French Indochina War, the Mexican Revolution and the Biafran War. These articles were not available in Greek.

We extracted the main text from which we removed the cross-page references, citation brackets and pictures. Then we placed the text inside the <*TEXT*> tag and modified the <*DCT*> (Document Creation Time) and <*DOCID*> (Name of the document) tags accordingly. The result was 19 files with the *.tml* extension. The entirety of these files consists of 76868 words. The completed *WikiWarsEL* corpus is publicly available on github[12].

3.1 Annotation Overview

The process of the annotation we followed was as close as possible to *TimeML Annotation Guidelines* [2] a part of which is inherited from *TIDES* [1] concerning the *TIMEX3* tag.

One Annotator read the texts at least twice and manually annotated temporal expressions with <*TIMEX3*> and its attributes *tid*, *type*, *value*, *mod* and *temporalFunction*. For each expression a decision was made on what extent the expression should be marked for annotation. This decision making process is summarized in Figure 3 flow chart. Afterwards, a unique value of the format "tX" was given to the *tid* attribute where “X*”* is an incremental integer. The value of the attribute *type* was selected accordingly from "DATE", "DURATION", "TIME" and "SET". Then the value of the attribute *value* was set according to ISO 8601 and the extensions of *TIDES*. If necessary, also the attributes *mod* and/or *quant* was set according to *TIDES*. Finally, the attribute *temporalFunction* was set to "true" when the value of the expression was under-specified or when the expression did not contain all the information necessary to fill the higher-order (left-hand) positions in the value and "false" in all other cases.

[11] https://el.wikipedia.org

[12] https://github.com/mkapernaros/WikiWarsEL

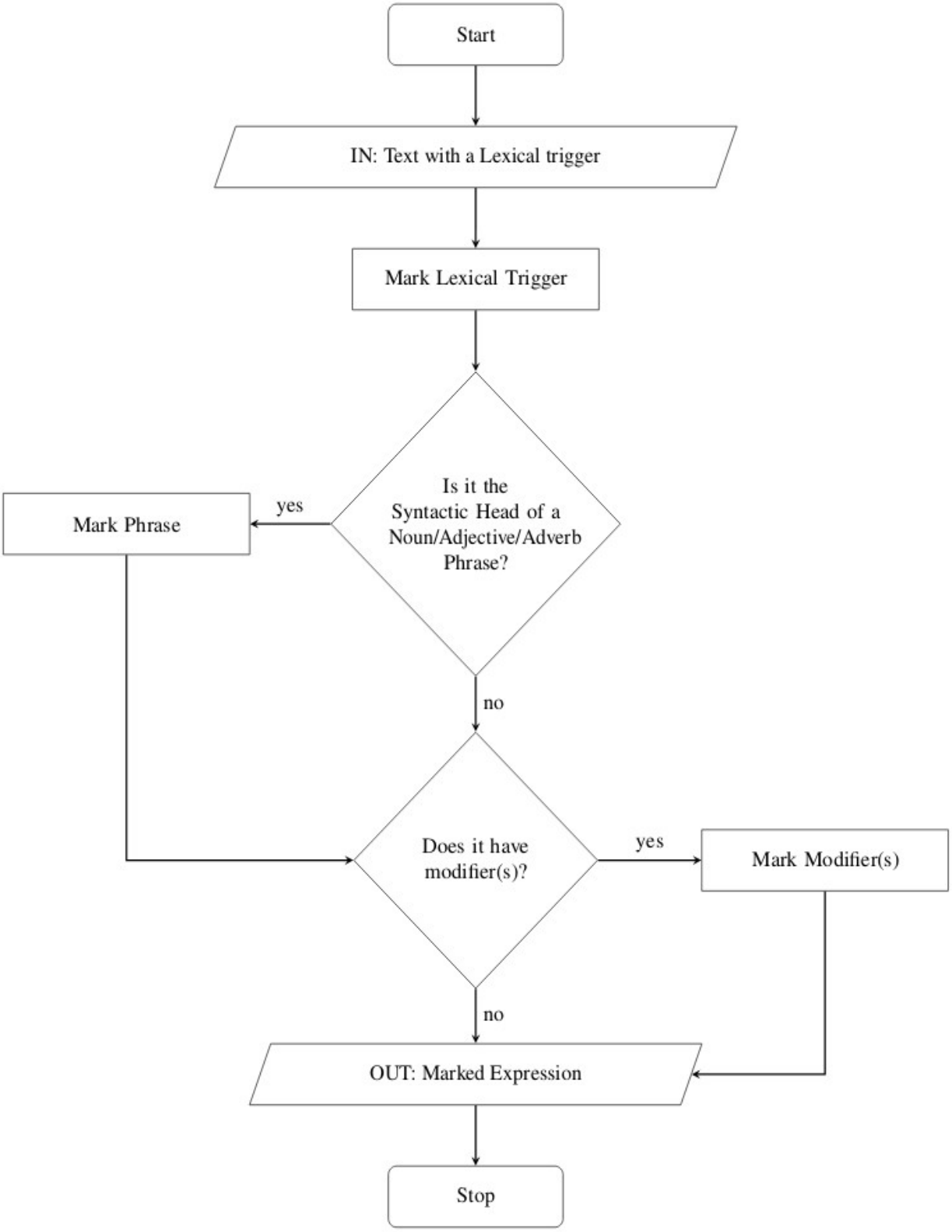

Figure 3: The process of Marking the expression to be Annotated

Here is an example expression with an english translation:

Ο στρατός αποχώρησε στο τέλος της ημέρας.
The army left in the end of the night.

Which is analyzed in:

- Lexical Trigger: "ημέρας"
- Noun Phrase: "της ημέρας"
- Modifier: "τέλος"

And here is after the annotation:

> Ο στρατός αποχώρησε στο <TIMEX3 tid="t129" type="DATE" value="1812-12-12" mod="END">τέλος της ημέρας</TIMEX3>

The development of *WikiWarsEL* was in parallel with the development of *Greek HeidelTime Resources* and this provided us the ability to detect some temporal expressions that the Annotator failed to notice while reading. Even though all articles were read at least two times by the Annotator, having an Automatic Tagger like *HeidelTime* was proven very useful for efficiency.

Note that on each of the two decision blocks in the flow chart, if the full marked expression was to become a prepositional phrase or a subordinating clause, then the answer was no and the additional text was not marked.

3.2 Conventions

Since the guidelines we followed have not been adapted for the Greek language, we had to make some conventions about whether or not an expression is mark-able.

3.2.1 Contractions

The guidelines dictate that the full extent of the annotated expression cannot be a prepositional phrase. That is it cannot start with a preposition. However, in Greek there is a phenomenon where an article and a preposition can be combined into one word. This is extremely frequent when referring to a time or date. For example, the English expression "*on June 17, 2020*" must be annotated excluding the preposition *on* but in the same Greek expression "*στις 17 Ιουνίου 2020*" the word *στις* is a contraction containing the preposition *σε* and the article *τις*. We made the convention not to include contractions in *WikiWarsEL*.

3.2.2 Compounds

On the other hand, we decided to include compounds when their head is a temporal lexical trigger and therefore denotes a duration as in "*διήμερο*" (two days).

3.2.3 Ages

The third convention we had to make was about ages like in "*17 ετών*" (17 years old). We decided not to include ages because they do not denote any point in time or a duration.

3.3 Annotation Examples

Here are some examples from *WikiWarsEL* using brackets to indicate what was annotated.

Example 1:

> Στις [30 Οκτωβρίου 1918] υπογράφεται η πρώτη ανακωχή του πολέμου
> In October 30, 1918 the first truce of the war is signed

This is the usual expression for a calendar date which starts with the contraction *Στις*.

Example 2:

> απέχοντας [μόνο λίγους μήνες] από τον αναπόφευκτο πλέον πόλεμο.
> being away only a few months from the inevitable war.

Here, both *μόνο* (adverb) and *λίγους* (adjective) are modifiers of the temporal *μήνες* hence their inclusion in the annotation.

Example 3:

> [ύστερα από περίπου εκατό χρόνια]
> about one hundred years later

This expression was annotated in its entirety even though it contains the preposition *από* because it is part of the adverbial modifier phrase "*ύστερα από*" which means "later". *Περίπου* (adverb) which means "about" is a second modifier of *εκατό χρόνια*.

For reference here are the same expressions when fully annotated:

Example 1:

> Στις <TIMEX3 tid="t149" type="DATE" value="1918-10-30"
> temporalFunction="false">30 Οκτωβρίου 1918</TIMEX3>
> υπογράφεται η πρώτη ανακωχή του πολέμου

Example 2:

> απέχοντας <TIMEX3 tid="t32" type="DURATION" value="PXM"
> temporalFunction="true">μόνο λίγους μήνες</TIMEX3> από τον
> αναπόφευκτο πλέον πόλεμο.

Example 3:

> <TIMEX3 tid="t17" type="DURATION" value="P100Y"
> mod="APPROX" temporalFunction="true">ύστερα από περίπου
> εκατό χρόνια</TIMEX3>

3.4 Statistics

The complete annotated corpus has 1575 temporal expressions tagged with *<TIMEX3>* which are further analyzed in Table 4.

Table 4: WikiWarsEL statistics by filename

Filename	Words	TIMEX	DATE	TIME	DURATION	SET
01_WW2.tml	7624	169	150	0	18	0
02_WW1.tml	6870	239	218	2	19	1
03_AmCivWar.tml	13468	161	128	7	27	0
04_AmRevWar.tml	1227	27	26	0	1	1

05_VietnamWar.tml	10489	233	186	7	36	5
06_KoreanWar.tml	2532	39	34	2	3	1
07_IraqWar.tml	518	22	23	0	0	0
08_FrenchRev.tml	5210	99	91	2	7	0
09_GrecoPersian.tml	6178	96	82	1	14	0
10_PunicWars.tml	2786	20	13	0	7	1
11_ChineseCivWar.tml	4080	72	67	0	6	0
12_IranIraq.tml	752	19	14	0	6	0
13_RussianCivWar.tml	7304	209	199	0	11	0
16_SpanishCivilWar.tml	2228	38	30	2	7	0
17_AlgerianWar.tml	1194	33	29	0	5	0
18_SovietsInAfghanistan.tml	1047	25	23	0	3	0
19_RussoJap.tml	1730	35	32	0	4	0
20_PolishSoviet.tml	435	12	13	0	0	0
22_SecondItaloAbyssinianWar.tml	1196	27	27	0	1	0
Total	76868	1575	1385	23	175	9

3.5 Notes and Further Work

One thing that recurred in the annotation process was finding expressions that the Annotator failed to detect when reading the texts even multiple times. A second Annotator may be needed for an independent evaluation of the annotation.

One factor that slowed down the process is the lack of Greek specific guidelines. We had to make conventions every time we found an expression that was not clear how to annotate. Sometimes this meant that we had to re-annotate the texts every time a new convention was made. Further work on adapting *TimeML* Guidelines to Greek may be helpful in future projects.

4. RESOURCES DEVELOPMENT

HeidelTime uses *.txt* files for language specific resources which can be created with any text editor following the syntax described in [4]. In this section we describe the development process of these *.txt* files which are divided in three categories. The *patterns* files, the *normalization* files and the *rules* files. This was a trial and error process involving development and testing, trying to match more temporal expressions from *WikiWarsEL* in each step. The final resources are publicly available on github[13].

4.1 Testing Automatic Resources

Before we begin discussing the development of the resource files, we can have an idea of the quality of *HeidelTime's auto-greek* resources. We evaluated *auto-greek* on the same test documents we used for the final evaluation which are not part of the training process and we present the results in Table 5. The results show that 60% of the extracted expressions are correctly extracted even though the extent of the expressions is 30% correctly matched. In other words Precision is 60% when counting relaxed matches and 30% when only counting strict matches in regards to the extent of the extracted expressions. The problem is that this is only a fraction (Recall is 2.75% and 1.38% accordingly) of all the temporal expressions in the test documents. *HeidelTime* with its automatic resources missed most of the expressions leading to Value F1-score of 2.19%. We will discuss in more depth these metrics in the final evaluation of our Greek resources in Section 4.4

Table 5: Auto-Greek Performance

	F-score	**Precision**	**Recall**
Extraction (Strict Match)	2.63%	30%	1.38%
Extraction (Relaxed Match)	5.26%	60%	2.75%
	Value	**Type**	
Normalization F1	2.19%	5.26%	

4.2 Extending Automatic Resources

4.2.1 Pattern and Normalization Resources

The first changes we made to *auto-greek* resources were to add month names, inflection cases and capital letters for the words in the beginning of a sentence. For example, two patterns were used for the word "*Ιανουάριος*" (January) to match Nominative, Genitive and Accusative cases:

```
// example for cases:
"Ιανουάριος?"
"Ιανουαρίου"
// example for capitals:
"[Σσ]ήμερα"
```

[13] https://github.com/mkapernaros/heideltime-resources-greek

Continuing on inflections, we started adding whenever needed genders and numbers to adjective, noun and article patterns. For example, for the adjective everyday in all numbers, cases, and genders the patterns are:

// english: everyday P1D
καθημεριν[όή]ς?
Καθημερινού
καθημερινοί
καθημερινές
καθημερινά
καθημερινών

We translated the weekday names in *normDayInWeek* which were in English and deleted some words which were in ancient Greek. We also changed the normalization values of duration according to *TIDES* and the normalization value for mid-day in *PartOfDay* according to *ISO 8601*. In particular we changed:

"DE" to "E" (for decades)
"CE" to "C" (for centuries)
"WE" to "W" (for weeks)
"L" (for "millennia)
"MD" to "MI" (for mid-day)

Additional pattern and normalization resource were developed including:

- ***reHour***, ***reMinute*** and ***normHourPM***, for time expressions like 5:00 pm to be normalized to T17:00
- ***PointsDurations***, for expressions which need the *mod* attribute when annotated
- ***AgoLater***, for modifier phrases
- ***PastFuture***, for phrases that denote the “PAST_REF”, “PRESENT_REF” and “FUTURE_REF” in the *value* attribute
- ***SetWord***, for expressions that denote a value when the *type* attribute is “SET”
- ***UnitWord4DurationPrefix***, ***UnitWord4DurationSuffix***, ***UnitTimeWord4DurationPrefix*** and ***UnitTimeWord4DurationSuffix***, for compound words like “*διετία”* (two years) and “*πεντάλεπτο*” (five minutes).

There are more additions and modifications which are not mentioned here. In terms of file and line changes Table 6 show the result of the commands *git diff --stat ./normalization/** and *git diff --stat ./repattern/** that we used to compare the respective folders of the Automatic Resources to the ones of the Manual Resources.

Table 6: Pattern an Normalization file changes

Folder name	Files changed	Line instertions	Line deletions
normalization	16	416	248
repattern	16	415	244

4.2.2 Rules Resources

The first rule we added had the purpose of matching four-digit years after the article "*το*" (the). This rule alone improved precision noticeably because this is a very frequent type of expression like "το 1940":

```
RULENAME="year_r1",
EXTRACTION="το %reYear4Digit",
NORM_VALUE="group(1)"
```

This rule, like every rule with an article in the beginning, was later changed to accommodate the fact that some words have a suffix identical to articles and invalidate the expression matching. This was solved with the boundary matcher "**\b**". Also instead of the article "*το"* we used the pattern file "*%reThe"* which contains all inflections of this article. So this rule became:

```
RULENAME="year_r1",
EXTRACTION="\b(%reThe)?%reYear4Digit",
NORM_VALUE="group(3)"
```

Making the article matching optional with the "**?**" quantifier in all rules turned out a good strategy to increase extent precision. Only one negative rule was necessary to discard 4-digit numbers which do not denote a year:

```
RULENAME="year_r1negative",
EXTRACTION="%reYear4Digit (μέτρα|εκατοστά|χιλιόμετρα|χλμ|δραχμές|δρχ|
δολάρια|$|ευρώ|€)",
NORM_VALUE="REMOVE"
```

We continued to add and modify rules but as the development was advancing we reached a point where every rule was essentially completely rewritten.

For convenience we organized the new rules in sections ordered by granularity. For example, we have separated the rules that match expressions of day granularity from the rules that match expressions of month granularity. The separation is done by the first word in their name. The rule name "*year_r1*" indicates that this rule is of year granularity. There are exceptions to this naming scheme but overall it was helpful when debugging. Here are two examples:

```
// EXAMPLE century_r1: 20ο αιώνα
// EXAMPLE setword_r1: καθημερινά
```

The first one matches expressions with century granularity like "*the 20th century*" and the second one matches set expressions like "*everyday*".

We developed 121 rules in 24 categories analyzed in Table 7.

Table 7: Developed rules count by type

	TOTAL	**DATE**	**DURATION**	**TIME**	**SET**
Automatically Developed	37	25	2	6	4

Manually Developed	121	69	17	26	9

4.3 Not covered expressions

There are still temporal expressions that are not detected (False Negatives) and expressions that are wrongly detected (False Positives). Even though the resources were developed with *WikiWarsEL* as the training set, they do not cover 100% of its expressions.

4.3.1 False Positives and False Negatives

We identified at least 3 categories of False Positive expressions for which we have not developed a sufficient rule set:

- **Idioms**: For example, the expression "*χρόνο*" (time/year) is matched even though it is part of the phrase "*κέρδισε χρόνο*" (bought time). We tried removing these expressions when they have a verb before the temporal unit using *POS_CONSTRAINT*. The result was 8 false positives to be removed, but with them there were 5 true positives also removed and we decided not to apply this approach.
- **Conjoined Expressions**: For example, "*το καλοκαίρι της ίδιας χρονιάς*" (the summer of the same year) is detected as two separate expressions instead of one. Additional rules that cover conjoined expressions with date and time were developed and worked successfully but we practically doubled the count of the respective rules. More work is needed to test this approach for the rest of the possible conjoined expressions.
- **Ages**: For example, "*ο 55χρονος*" (the 55 years old man) which is exactly the same as the duration of 55 years. We did not test any solution for these expressions.

The situation with False Negatives is different. There are some types of expressions that we failed to match with regular expressions.

- **Dots**: For example "*το 494 π.Χ.*" (494 B.C.) could not be matched by creating a pattern like "*το \d\d\d π[\.]Χ[\.]*". The same applies to time expressions when they use a dot instead of a colon to separate hours from minutes. This kind of expressions is the majority of False Negative expressions.
- **Semicolon and Comma**: For example when the expression ends with a semicolon (·) or a comma, the expressions are not matched. No solution was tested.
- **Numbers**: For example when a number denotes a day of a month but it is not accompanied with some other lexical trigger. We did not develop any rules for these cases.

4.3.2 Relaxed matches

These are the expressions that were matched but not in their entirety concerning their extent. Two categories of such cases are the following:

- **Multiple Modifiers**: For example, the expression "*ύστερα από περίπου εκατό χρόνια*" (about one hundred years later) have two modifiers *ύστερα* and *περίπου*

which could be also in different positions. For such cases we have not developed any rules.

- **Variety of Modifiers**: For example, "*15 ολόκληρα χρόνια*" (15 whole years), "*την άλλη μέρα*" (the other day) and "*νέο έτος*" (new year) contain adjectives that were unusual and we did not develop any rules for such cases.

4.3.3 Resources Syntax

For the most part the syntax is straightforward and we immediately managed to adapt and expand *auto-greek*. During the development we noticed two cases that we did not expect.

When we use *UNDEF-year* or *UNDEF-month* for the whole value, without the *-REST* part, the disambiguation phase returns unexpected results like the *DCT year* even though *HeidelTime* is set to narrative article type. If instead we use *UNDEF-this-year*, then the problem for the most part is resolved. This behavior is consistent for also century, and day units.

Another problem we encountered was with the use of *OFFSET* and the "**?**" quantifier in the same rule. If the group that is quantified like the article *το* in "(το)? (\d\d\d\d) - (\d\d\d\d)" does not exist, then we cannot extract the groups using their expected numbers, for example with "OFFSET="group(3)-group(3)*""*. We solved this by creating two separate rules for such expressions, one with the article and the other without the article.

4.4 Evaluation

For the evaluation of our manually developed resources we used 3 test document files created similarly to the document files in *WikiWarsEL*. The document files are *February_Revolution.tml*, *Greek_Civil_War.tml* and *Napoleonic_Wars.tml* which contain the relevant war text from the Greek *Wikipedia*. In total there are 18 thousand words in these test files. The test files are available at github[14].

The metrics we will use to quantify the performance/quality of the resources are Precision, Recall and F-score. In Figure 4 we define these terms where TP, TN, FP and FN stand for True Positives, True Negatives, False Positives and False Negatives accordingly.

$$Precision: P = \frac{TP}{TP+FP}$$

$$Recall: R = \frac{TP}{TP+FN}$$

$$F_1 - score: F_1 = \frac{2*P*R}{P+R}$$

Figure 4: Statistical metrics definitions

We can describe Precision as the percentage of the correctly extracted expressions out of all the extracted expressions. Similarly, Recall is the percentage of the correctly extracted expressions out of all the expressions which should have been extracted. We calculate Precision and Recall both for Strict Matches and Relaxed Matches. A Strict Match is defined as the extraction of the complete extent of an expression and a Relaxed Match as the extraction of at least one character of an expression.

[14] https://github.com/mkapernaros/TestWarsEL

With F-score we can have a metric that takes into account both Precision and Recall and in our case this is 85.87% for the Strict Match Extraction Phase and 94.33% for the Relaxed Match Extraction Phase as presented in Table 8. Incorporating the results of the *Normalization Phase* into the F-score, where TP this time is the correctly extracted expressions with correct normalization values, we receive 82.31% Normalization F1.

Table 8: Greek Resources Performance

	F-score	**Precision**	**Recall**
Extraction (Strict Match)	85.87%	83.37%	88.53%
Extraction (Relaxed Match)	94.33%	91.58%	97.25%
	Value	**Type**	
Normalization F1	82.31%	90.32%	

Table 9: All performance test results

Tests	**Strict Match**			**Relaxed Match**			**Normalization F1**	
	F-score	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**Value**	**Type**
auto-greek on WikiWarsEL	8%	65.05%	4.26%	11.69%	95.15%	6.23%	6.09%	10.14%
auto-greek on TestWarsEL	2.63%	30%	1.38%	5.26%	60%	2.75%	2.19%	5.26%
new-greek on WikiWarsEL	87.72%	87.78%	87.67%	96.25%	96.31%	96.19%	82.32%	91.09%
new-greek on TestWarsEL	85.87%	83.37%	88.53%	94.33%	91.58%	97.25%	82.31%	90.32%

5. CONCLUSION

In this work we extended the Greek language support in HeidelTime. In the beggining we tested HeidelTime on the Greek documents *TestWarsEL* and the results were of low performance with F1-score of 2.19%. This performance could be easily improved with the development of hand-crafted language specific resources that were not available and instead only automatically generated resources existed for the Greek language.

HeidelTime is designed to be extendable with more resources which are separate from its source code. We examined the automatically generated Greek resources that were available and found many areas that could be improved. All sorts of changes were necessary from deleting obsolete words like ancient Greek words, to syntactic corrections and more importantly additions of missing words and rules for many expressions. A basic level of familiarity with regular expressions was a prerequisite but apart from that the resources were easy to understand and to extend them. In addition there are already available various scripts available that made the debugging process smoother. We could effectively detect false positives and false negatives during the development which guided the way to completing our task. Some of the resources were kept with minimal changes but the vast majority were written from scratch.

The development was practically impossible without having a corpus already annotated to use it as a training set. Without a corpus we could not test our progress as we would not with certainty the effect of the newly developed resources. For this purpose we started also the development for a new Greek annotated corpus. We developed the *WikiWarsEL* corpus inspired by the English *WikiWars* and its *German* and *Vietnamese* versions. This corpus was carefully annotated by one Annotator with the tag *<TIMEX3>* for every temporal expression that was found in the text. We followed the *TimeML* guidelines to annotate the text with at least two passes for every text file to lower the probability of missing some temporal expressions.

After the development reached a level of high performance we froze any further work on the resources and instead we focused on the evaluation process. So we ended up producing the newly created Greek resources and the newly created *WikiWarsEL* corpus. We wanted to evaluate the performance of the new resources on Greek text documents that were not used on the training process and did not interfere with the development process at all. This was the time to create *TestWarsEL*, a collection of Greek war documents, that we used for the evaluation process. After this point we did not do any further development on the resources to avoid biased modifications towards the *TestWarsEL* documents.

The evaluation process showed an increased quality of our new resources overall over the automatically generated resources. In particular, the biggest weakness of the automatic resources is noted in the Recall metric with the value of 2.75% which means that only 2.75% of the temporal expressions inside the test documents were detected. This number is for the *Relaxed Matches* where even if only one character of an expression is correctly matched, it counts as a *True Positive*. The second weakness was that only 60% (Precision 60%) of the expressions that were detected were truly temporal expressions which ultimately resulted in the very low Extraction F1-score of 5.26% and when we take into account the normalization correctness we end up at the even lower F1-score 2.19%.

All of the above weaknesses were remedied with our manually developed resources as we can observe on the evaluation process. Precision and Recall went up to 91.58% and 97.25% respectively resulting in Extraction F1-score of 94.33% and when we take into

account the normalization correctness we end up with F1-score 82.31% which is comparable to previous works on different languages.

It is shown that HeidelTime performs Temporal Tagging of high quality for the Greek language similarly to what is shown in previous works for Arabic, Italian, Spanish, Vietnamese[3] and Chinese[8]. The effort of developing Greek language specific resources was minimal because of the simplistic syntax and file structure which also makes them suitable for quick and easy updates or revisions. The evaluation on test documents that were not used in the training processes resulted an 82.31% F1-score even though some expression categories were left out of training. For the annotation task of temporal expressions in formally written documents, this approach can be used effectively. More work is needed for the adaptation of the Annotation Guidelines for the Greek language and to investigate for possible solutions to improve the newly developed Greek Resources.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	**Ελληνικός Όρος**
Extraction	Εξαγωγή
Normalization	Κανονικοποίηση
Temporal Tagging	Χρονικός Σχολιασμός
Regular Expresssion	Κανονική Έκφραση
Metadata	Μεταδεδομένα
Corpus	Σώμα
Training Set	Σύνολο Εκπαίδευσης
Value	Τιμή
Evaluation	Αξιολόγηση
Standard	Τυποποιημένη

ABBREVIATIONS - ACRONYMS

TP	True Positives
FP	False Positives
TN	True Negatives
FN	False Negatives

REFERENCES

[1] L. Ferro, L. Gerber, I. Mani, B. Sundheim, and G. Wilson, "Instruction manual for the annotation of temporal expressions," *MITRE Technical Report MTR 01W0000046,* 2001.

[2] R. Saurı, J. Littman, B. Knippen, R. Gaizauskas, A. Setzer, and J. Pustejovsky, "TimeML Annotation Guidelines Version 1.2. 1," 2006.

[3] J. Strötgen, A. Armiti, T. Van Canh, J. Zell, and M. Gertz, "Time for more languages: Temporal tagging of Arabic, Italian, Spanish, and Vietnamese," *ACM Transactions on Asian Language Information Processing (TALIP)* 13, no. 1: 1-21, 2014.

[4] J. Strötgen, and M. Gertz, "Multilingual and cross-domain temporal tagging," *Language Resources and Evaluation* 47, no. 2: 269-298, 2013.

[5] J. Strötgen, A. Armiti, T. Van Canh, J. Zell, and M. Gertz, "Time for more languages: Temporal tagging of Arabic, Italian, Spanish, and Vietnamese," *ACM Transactions on Asian Language Information Processing (TALIP)* 13, no. 1: 1-21, 2014.

[6] J. Strötgen, and M. Gertz, "WikiWarsDE: A German corpus of narratives annotated with temporal expressions," in *Proceedings of the conference of the German society for computational linguistics and language technology (GSCL 2011)*, pp. 129-134, 2011.

[7] P. Mazur, and R. Dale, "Wikiwars: A new corpus for research on temporal expressions," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 913-922. Association for Computational Linguistics, 2010.

[8] H. Li, J. Strötgen, J. Zell, and M. Gertz, “Chinese temporal tagging with HeidelTime”, in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers* (pp. 133-137), 2014.