The INFOCORE Dictionary

A multilingual dictionary for automatically analyzing conflict-related discourse

Principal Author:

Christian Baden, The Hebrew University of Jerusalem (ontology, lead for English, validation, syntax, techn. realization)

Co-Authors:

Marc Jungblut, Ludwig Maximilian University Munich (ontology, lead for German/French)
Igor Micevski, RESIS Skopje (ontology, lead for Albanian/Macedonian/Serbian)
Katsiaryna Stalpouskaya, gutefrage.de (technical realization)
Keren Tenenboim-Weinblatt, The Hebrew University of Jerusalem (ontology, lead for Hebrew/Arabic; WP leader, co-owner)
Rosa Berganza Conde, King Juan Carlos University Madrid (WP leader, co-owner)
Dimitra Dimitrakopoulou, Massachusetts Institute of Technology (WP leader, co-owner)
Romy Fröhlich, Ludwig Maximilian University Munich (project coordinator, WP leader, co-owner)

Other Contributors:

Tali Aharoni (ontology, Hebrew), Maéva Clément (French), Anke Fiedler (ontology), Patricia Gautier (French), Yonatan Gonen (Arabic, Hebrew), Beatriz Herrero (ontology), Bojan Ilijoski (Serbian), Yuval Katz (Hebrew), Cristina Monzer (German, English), Asmahan Simry (Arabic)

Recommended Attribution:

Baden, C., Jungblut, M., Micevski, I., Stalpouskaya, K., Tenenboim-Weinblatt, K. Berganza Conde, R., Dimitrakopoulou, D., & Fröhlich, R. (2018). *The INFOCORE Dictionary*. Available Online: https://osf.io/f5u8h/ or https://github.com/christianbaden/INFOCORE

This dictionary was created as part of the **INFOCORE research project**, coordinated by *Romy Fröhlich, Ludwig Maximilian University Munich*; as part of the work of the **Methodological Working Group: Content Analysis**, headed by *Christian Baden, The Hebrew University of Jerusalem;* which was a joint effort of *Work Package 5: Social Media*, headed by *Dimitra Dimitrakopoulou, Massachusetts Institute of Technology*; **Work Package 6: Strategic Communication**, headed by *Romy Fröhlich, Ludwig Maximilian University Munich*; **Work Package 7: Journalistic Transformation**, headed by *Keren Tenenboim-Weinblatt, The Hebrew University of Jerusalem*; and **Work Package 8: Parliamentary Discourse**, headed by *Rosa Berganza Conde, King Juan Carlos University Madrid.*

The INFOCORE Dictionary includes…

- 3,738 measured concepts in conflict-related discourse, ordered into four main groups: Semantic concepts (IDs 1####); Actors (2####); Times/Events (3####); Locations (4####)
- 8 languages: Albanian, Arabic, English, French, German, Hebrew, Macedonian, and Serbian
- 78,984 unique search phrases with 191,408 disambiguation criteria (Albanian: 9,595 + 22,915; Arabic: 9,592 + 10,816; English: 10,464 + 38,459; French: 7,476 + 19,470; German: 14,054 + 46,208; Hebrew: 11,708 + 14,853; Macedonian: 7,727 + 18,450; Serbian: 8,368 + 20,237)

The INFOCORE Dictionary is built for use in conjunction with the jamcode coding script available on Github, and the AmCAT open source text analysis platform.

I. Purpose & Application

The INFOCORE Dictionary was created as part of the EU-funded collaborative research project INFOCORE ("(In)forming Conflict Prevention, Response, and Resolution: The Role of Media in Violent Conflict"; Grant Agreement No. 613308, 1.1.2014 - 31.12.2016, Coordinator: Romy Fröhlich, Ludwig Maximilian University Munich). The project aimed to investigate the role(s) that media play in the emergence or prevention, the escalation or de-escalation, the management, resolution, and reconciliation of violent conflict. For this purpose, INFOCORE aimed to provide a systematically comparative assessment of various kinds of media, interacting with a wide range of relevant actors and producing diverse kinds of conflict coverage. It focused on six different violent conflicts located within three main conflict regions – the Israeli-Palestinian conflict and the Syrian civil war in the Middle East; the conflict in Kosovo, and ethnic tensions in (now: Northern) Macedonia in the West Balkans; and the various conflicts in Burundi and the Democratic Republic of the Congo in the African Great Lakes area. Its findings address both the socially interactive production process behind the creation of conflict coverage, and the dynamics of information and meaning disseminated via the media.

Specifically, INFOCORE focused on the conditions that bring about different media roles in the cycle of conflict and peace building. It generated knowledge on the social processes underlying the production of conflict news, and the inherent dynamics of conflict news contents, in a systematically comparative fashion. Based on this perspective, the project identified conditions under which media play specific constructive or destructive roles in preventing, managing, and resolving violent conflict, and building sustainable peace. To do so, INFOCORE reconstructed the production process of conflict-related media contents, focusing on the interactions between professional journalists, political actors, experts/NGOs, and lay publics. It analyzed these actors' different roles as sources or advocates, mediators, users and audiences in the production of professional news media, social media, and semi-public expert analysis. To assess the roles of media for shaping conflict perceptions and responses to ongoing conflicts, INFOCORE analyzed the dynamics of conflict news content, and its interaction with other key domains of public discourse – notably, strategic communication, parliamentary discourse, and social media – over time. It identified recurrent patterns of information diffusion and the polarization/consolidation of specific frames and contextual factors that influence the roles media play in conflict and peace building.

INFOCORE was overall divided into two groups of work packages. Four work packages:
Work Package 1: Journalists; PI: *Thomas Hanitzsch, Ludwig Maximilian University Munich*
Work Package 2: Political Actors; PI: *Gadi Wolfsfeld, Interdisciplinary Center Herzliya (now: Reichman University)*
Work Package 3: Lay Publics; PIs: *Marie-Soleil Frere, Free University Brussels, & Snezana Trpevska, School for Journalism & Public Relations Skopje*
Work Package 4: Experts; PI: *Christoph O. Meyer, King's College London*
…focused on interview-based methods to reconstruct the processes and actors' perspectives upon the production and reception of conflict-related discourse.

By contrast, the four other work packages:
Work Package 5: Social media; PI: *Dimitra Dimitrakopoulou, then at the Hellenic Foundation of European & Foreign Policy ELIAMEP)*
Work Package 6: Strategic communication; PI: *Romy Fröhlich, Ludwig Maximilian University Munich*
Work Package 7: Journalistic Transformation; PI: *Keren Tenenboim-Weinblatt, The Hebrew University of Jerusalem*
Work Package 8: Parliamentary Discourse; PI: *Rosa Berganza Conde, King Juan Carlos University Madrid*
…focused on content analytic methods to model and analyze the dynamic process of conflict news dissemination and evolving public discourse.

The INFOCORE Dictionary was created by the Methodological Working Group: Content Analysis, a joint effort of Work Packages 5, 6, 7, and 8, led by *Christian Baden, The Hebrew University of Jerusalem.*

Within the context of this project, the primary purpose of the INFOCORE Dictionary was to enable a systematic comparative analysis of the ways in which conflicts are discussed…

- across different conflicts
- across different languages
- across different countries
- across different arenas
- over time

Moreover, the dictionary was intended to capture the content of conflict-related debates at a low level of abstraction, to permit an inductive identification of recurrent patterns, as well as qualitative differences, in fine grain. At the same time, the analysis needed to cover immense amounts of data produced by strategic communicators (e.g., in the form of press releases, statements), journalists (across different media genres), political actors and institutions in political and parliamentary discourse, and active lay publics (on social media).

For this purpose, a high-resolution multi-lingual dictionary was chosen as the approach of choice, owing to several key advantages:

- Scalability: A dictionary can be applied to any number of documents without additional costs
- Detailed research control: A dictionary permitted us to diagnose and adjust operational decisions at every stage, facilitating thorough manual validation and adjustment
- Transparency: A dictionary enables anyone to scrutinize and follow how cross-linguistic and cross-context equivalence was achieved for each measured concept
- Support for fine grain: A dictionary permitted us to include as many concepts as necessary to capture important differences in the debate.

The governing quality criteria for devising the dictionary were:

- that the information included in any relevant document can be reconstructed from the sequence of coded concepts
- that the same concept expresses commensurable meaning across all kinds of debates and contexts.[1]

The dictionary was created with a specific focus on conflict-related discourse in social media, strategic communication, journalistic news, and parliamentary discourse, both inside (Israel, Palestinian Territories, Syria, Kosovo, Macedonia, Burundi, DR Congo) and outside the conflict areas themselves (in Germany, France, Spain, UK, and to some extent U.S. and transnational news agencies), over an extended period of time. At the same time, the dictionary is tailored to the specific conflicts in question, foregrounding comparative dimensions that matter for comparing discourse regarding these conflicts and possibly others, in the time period between 2006 and 2014. It may still apply to other discourses elsewhere, and at different times, but it will require re-validation of relevant key categories, and most likely some extent of adaptation.

For additional background on INFOCORE, please consult the project webpage, www.infocore.eu, which lists a wide variety of project-related publications and other outcomes. Parts of this documentation were already part of INFOCORE Working Paper 2015/10: Common Methodological Framework: Content Analysis: A mixed-methods strategy for comparatively, diachronically analyzing conflict discourse, authored by Christian Baden and Katsiaryna Stalpouskaya.

[1] By commensurable, we do not necessarily mean that expressions must be semantically equivalent, but rather, that they serve the same pragmatic, semantic or symbolic function in the discussion of violent conflict. For instance, we grouped references to several known liberal policies (on soft drugs, abortion, etc.) because all of these were used consistently in the observed debates to refer to an ideal of very liberal societies.

II. Dictionary Creation Process

The INFOCORE dictionary is the outcome of a long, multi-stage research process, which proceeded in three main steps:

1. *Inductive ontology creation*

In a first step, the INFOCORE team departed from a qualitative reading of a wide range of documents all across the intended data sample. Documents were sampled to ensure variability across…

- text genres (social media, strategic communication, news coverage, parliamentary debates) and provenance (social media platform, strategic actor, news outlet, government/opposition)
- languages (Albanian, Arabic, English, French, German, Hebrew, Macedonian, Serbian)
- conflict relevance (Israel-Palestine, Syria, Kosovo, Macedonia, Burundi, DR Congo)
- time (up to 9 years, from 2006 and 2014, depending on the conflict in question)

Documents were sampled following a theoretical sampling logic, aiming to add different kinds of documents and seeking difference. Documents were initially analyzed by separate teams of native and highly proficient speakers of each language, but progress was synchronized over time between languages.

To construct an initial ontology for the dictionary, the qualitative stage proceeded as follows:

1.1. *Open reading, annotation, & recording.* As a first step, sampled texts were read with the purpose of identifying all meaning carrying concepts whose omission of replacement in the text would alter the meaning constructed. The underlying purpose was to identify any kinds of actors, objects, places, activities, processes, qualities, temporal and event references that needed to be captured to faithfully summarize the information contained in each document. This yielded a corpus of texts with highlighted contents, and a very long list of words or expressions that had been found important to record.

1.2. *Conceptual abstraction & ordering.* Next, the identified words were mapped onto semantic concepts. This involved a) a process of abstraction (what does a word express that makes it relevant to understand the meaning of a text; the conceptual meaning expressed is then added as an annotation to the highlighted corpus from step 1); b) a process of instantiation (what other words could be inserted that have equivalent meaning in the sense considered relevant); and c) a process of ordering the concepts: Some concepts are parents or children of other concepts (e.g., “Operation Protective Edge” is one possible child of a more abstract concept “Israeli limited military operations in Gaza”, under which “Molten Lead” and “Pillar of Defense” would be other children; and the parent could in turn be subordinated to a wider, internationally transferrable conceptual category of “limited military operations”). In the process of hierarchical ordering, some levels of abstraction could already be discarded if it was already clear that they would not be suitable for comparative analysis (too concrete or too abstract). As there are often multiple ways for constructing conceptual hierarchies, the guiding principle applied here was to group phenomena that are also described jointly, or likened to one another, in actual discourse. Also, concepts could be ordered with regard to wider semantic domains (e.g., actors, places, actions), their semantic or pragmatic quality (e.g., distinguishing described performatives such as “X threatens” from performed ones such as “if these attacks do not stop, we will…” and also from related assertions such as “the threat”), or other categorical differences.

With regard to the kinds of concepts covered, the main criteria for inclusion were:

- Relevance to understanding the meaning of the conflict or important conflict aspect discussed
- Medium level of generality: neither too specific (e.g., it makes little sense to distinguish “abhorrent” from “barbaric” if discourse participants use these interchangeably, or it makes no difference for the meaning constructed) nor too general (e.g., subsuming Gaza and the Westbank under one common concept loses some distinctions that are very relevant for understanding the conflict)

- “Translatability” into other languages, conflicts, and contexts

2. *Conceptual integration of the ontology*

2.1. *Comparative integration & definition*. While the above two stages operated primarily on sets of texts taken from specific kinds of outlets, countries and contexts, and were conducted separately by native speakers of the relevant languages, the construction of one unified ontology required integrating these separate lists. Once saturation was near in the above construction stage, the list of concepts identified in the different contexts were merged. This process started with a back-and-forth translation of concepts and their respective indicators between languages. Translators were instructed to focus on any common ways of expressing the same meaning in the respective other languages, while remaining faithful to connotations or semantic differences between related concepts. In this process, equivalent concepts were mapped where possible; where concepts were related but not equivalent, integration normally implied a shift in the level of abstraction, subsuming both under a common, more abstract construct, and completing the missing, more specific concepts in the respective other original contexts; concepts found in some but not in other contexts were retained unless irrelevant for comparative analysis, and suitable expressions to express the same meaning in the different context were identified. In this process, definitions were laid down for a large number of conceptual categories, elucidating which kinds of more specific meanings expressed were to be subsumed under these categories, and identifying the hierarchy of nested concepts.

2.2. *Empirical & logical completion*. Based on the master list, the respective original discourse samples were revisited in order to identify missing concepts that became salient only once the concepts relevant in other contexts were known (empirical completion). This stage required close collaboration between the teams working on the different text samples. The last step toward the final concept list was logical completion, where concepts that were not actually found in the analyzed texts but constitute a logical possibility were added. The logical completion specifically drew closely on the definitions of existing conceptual categories.

3. *Empirical Completion & Validation*

3.1. *Indicator completion*. Once the concept list was approximately complete, the list of indicators for each concept was revisited to find additional expressions that might occur. Part of this completion still followed the qualitative and logical procedures detailed above (e.g., using thesauri to identify synonyms or spelling variants; using translations from the indicators used for different languages or contexts). The procedure also included paying attention to the fact that some concept references are location sensitive (e.g., British media most likely refer to the German government as “German government” or using metonymic references such as “Germany”, “Berlin”, or “Merkel”; however, neither is true in the German media, which would just say “government” but use “Berlin” as shorthand for the German government only under specific, narrowly constrained circumstances). The main part of indicator completion consisted in the quantitative application of the draft dictionary to large and heterogeneous corpora of texts in the respective languages (i.e., shifting from the qualitative subsample used in step 1 to random subsamples of the full INFOCORE data collection, which comprised more than 6.5 million unique documents across all genres, languages, and conflicts). For indicator completion, we exported annotated texts, wherein every recognized concept was highlighted and labeled with the recognized concept label. A large team of research assistants then determined what additional expressions should have been recognized under the given ontology, proposing additional indicators for each language, which were then added and evaluated. This procedure was continued until barely any new indicators were turned up by subsequent rounds of validation, and recall was in an acceptable range.

3.2. *Disambiguation*. For each indicator, we furthermore ascertained that uses of this word reliably and validly express the relevant conceptual meaning. For all words that can be used in different senses or contexts, this required the construction of disambiguation rules that differentiate relevant from irrelevant uses.

Disambiguation rules were found partly in the annotated text format described above, but mainly through keyword-in-context (KWIC) searches in the population of texts. Most disambiguation criteria identified other words that must, or must not be present nearby a detected keyword to justify the conclusion that the keyword is a valid reference to a specific conceptual category. Other disambiguation strategies drew upon available meta-data (e.g., recognizing a name as reference to a specific office between the date of inauguration and the date of relief; utilizing specific language conventions typical for a venue or kind of text only in texts of a certain kind). The formulation of disambiguation criteria also involved validating a suitable distance wherein additional words had to appear (or not appear), balancing over-inclusive criteria (which respond to too many irrelevant mentions) versus over-restrictive ones (which miss valid mentions). This procedure was repeated many times, until precision was in an acceptable range.[2]

3.3. *Validation*. For formal validation, the final dictionary was applied to a random subset of the entire INFOCORE corpus. Based on the returned coding decisions, the precision (ratio of correctly recognized instances over all recognized instances) and recall (ratio of correctly recognized instances over all relevant instances; recall is tested based on the annotated text corpus constructed in steps 1 and 2, as well as a manual inspection of a random sample of the automatically coded texts) was evaluated. The final scores were:

Language	Precision	Recall
Albanian (AL)	0.507	0.716
Arabic (AR)	0.953	0.962
English (EN)	0.939	0.963
French (FR)	0.855	0.961
German (DE)	0.930	0.917
Hebrew (HE)	0.764	0.823
Macedonian (MA)		
Serbian (SR)	0.572	0.655

Please note:
The obtained precision and recall scores were obtained based on a wholistic validation of documents, not separately for each concept. As a consequence, scores depend on the relative frequency of concepts (i.e., frequent concepts that are less accurately recognized depress scores more than infrequent concepts). To interpret these scores, precision expresses how many of all concepts automatically labeled in a document were correct; and recall expresses how many of all concepts referenced in a document were correctly labeled.
It is thus likely that some (especially rare) concepts still perform badly, or even include mistakes (we occasionally still find typos in the tens of thousands of search phrases).

[2] Resources for validation were a bit unevenly distributed, with the effect that more validation rounds were conducted for the English, German, French, Hebrew and Arabic dictionaries, and fewer for the other three. As a result, validation scores for Macedonian are missing, and performance is notably weaker for Albanian and Serbian. The diminished performance for Hebrew has to do with the structure of Hebrew language, which creates large numbers of homographs that are excessively difficult to anticipate and disambiguate.

III. Syntax

The dictionary uses a syntax designed for use in conjunction with the jamcode coding script. This syntax…

- permits operationalizing each construct using multiple keywords, supporting truncation wildcards
- permits using auxiliary disambiguation criteria attached to each keyword, which specify which other strings must or must not be found withing a given distance of a recognized keyword; which pre- or suffices can be attached to the keyword; or on what dates the keyword is found
- permits restricting the applicability of keywords by date range

The syntax works as follows.

Any concept is operationalized using one or multiple search phrases. Search phrases are separated by space: Everything that is between two spaces is considered to be part of the same search phrase, and everything that is separated by spaces is considered to be part of distinct search phrases.

Please note:
Unnecessary spaces are the most common source of error. Under all circumstances, avoid double spaces, since this creates an empty search phrase (everything between two spaces) that will match any location in a document.

Every search phrase begins with a keyword. A keyword can be any character string that consists of letters in any recognized alphabet or script, numbers, or emojis. Punctuation marks cannot be part of keywords, and also selected special characters are excluded (notably, these: ()_&|*~). Keywords can be truncated, using * as truncation mark. Truncations can be at the beginning ("*abc" will match "abc", but also "xyzabc") and/or at the end ("abc*" will match "abc", but also "abcdef") of a keyword, but not in between ("a*c" will not work). If no truncation is defined, keywords match only character sequences enclosed by spaces (e.g., "abc" will not match "abcd", but "abc*" will; the exception to this rule is that in Hebrew and Arabic, certain common prefixes and suffixes will be automatically included, see below).

To use additional disambiguation criteria, the keyword is followed (without intermittent space) by an underscore _, an identifier of the type of disambiguation criterion (t,y,n,p,s), and a bracket that defines the criterion (e.g., "abc*_y(…)". The same keyword can be amended by any number of disambiguation criteria. The keyword will be matched only if all disambiguation criteria are fulfilled (e.g., "*abc_y(…)_n(…)_n(…)" will be matched only if all three criteria are met).

Please note:
Disambiguation criteria are optional. Everything within each disambiguation criterion must be entered without any spaces (everything separated by spaces will be considered to be part of a different search phrase). Multiple criteria are concatenated using underscores.

There are five types of disambiguation criteria, which fall into three classes: temporal criteria, context words, and affixes.

1. *Temporal criteria*

Keywords can be disambiguated based on document metadata, notably, the registered publication date. This permits, for instance, coding office holders only while they hold office (so "Trump" is coded as a reference to "US President" only while he is in office, but not before or afterwards).

_t(…)		This type of criteria defines the date from and until when a keyword is considered.

Temporal criteria are specified as dd/mm/yy-dd/mm/yy, using double digits for single-digit days, months and years; years are assumed to be in the 2000s, i.e., _t(03/07/14-10/07/14) refers to 3-10 July 2014. Start date and end date can be identical (so only one day will be considered), but the end date cannot be before the start date.

<u>*Please note:*</u>
Unlike the date criterion in the dictionary line (see below), this criterion governs only the use of a specific search phrase, while all other search phrases for the same concept will still be applied.

2. Context words

Keywords can be disambiguated based on whether other character sequences are found within a given word distance.

_y(…) This type of criteria defines which additional character sequences *must* be found for a keyword to be matched.

_n(…) This type of criteria defines which additional character sequences *must not* be found for a keyword to be matched.

Context words criteria contain two types of information: The character sequences that must or must not be found; and the word distance wherein the criterion applies. Character sequences are listed at the beginning of the bracket, using the following Boolean operators:

&	AND
|	OR
()	brackets

Character sequences are defined in the same way as keywords, i.e., they can contain letters, numbers, and emojis, and can use truncation, but cannot include punctuation or excluded special characters.

The word distance is defined at the end of the bracket, by a number preceded by ~. Word distance is symmetric, i.e., ~5 denotes a word distance of up to 5 words before or after a detected keyword. Word distance automatically recognizes the beginning or end of documents (so if the document ends within the specified distance, only existing words are considered)

<u>*Please note:*</u>
Within the jamcode *coding script, punctuation is automatically converted into word distance: Minor syntactic breaks (commas, semicola, dashes) are counted as one word; sentence breaks (periods, question marks, exclamation points) are counted as three words; and paragraph breaks are counted as five words. Thus, in the example "abc de fgh i. jkl, mno pq.", the distance between "abc" and "pq" is counted as 10 words (three words, three for the period, one word, one for the comma, two words).*

For instance, the disambiguation criterion _y(abc|def*~5) is matched if either "abc" or "def*" (or both) are found between five words before and five words after the detected keyword; the disambiguation criterion _n(*abc*|(def&gh)~10) is matched if the ten words before and after the detected keyword include neither a word that contains the "abc", nor both of the sequences "def" and "gh"; if said words contain only "def", the criterion is not violated (it excludes only the joint use of "def" and "gh"), so the keyword would be matched.

<u>*Please note:*</u>
Each layer of brackets must contain only operands of the same type; for instance, (a|b|(c&d)|e~5) is consistent, as is (a&(b|(c&d))~10) or ((a&b)|(c&d)|(e|f)), although the last bracket around "e" and "f" is unnecessary; By contrast, (a&b|c~15) is inconsistent. Whenever operands of different kinds coexist in a criterion, they need to be ordered by consistent brackets;
A bracket within the criterion must always include at least two elements; _y(abc~2) is consistent, but _y((abc)~2) is not, nor is _y(a|b|(c)~2);
The current coding script supports only up to five levels of nested brackets; Unbalanced brackets will raise an error.

3. ***Affixes***

Keywords can furthermore be disambiguated based on what prefixes and suffixes are permitted. This type of criteria is relevant under two conditions:

1) for truncated keywords, it excludes specific characters in the truncation;

For truncated keywords, it is sometimes useful to exclude specific characters, so the keyword can be preceded or followed by any character except for those specified (e.g., “aid*” also matches “AIDS”, but if the suffix “s” is excluded, it does not)). The character position checked for consistency with this criterion is the one immediately adjacent to the character sequence without the truncation.

2) for Arabic and Hebrew language;

Arabic and Hebrew both tend to affix various function words to keywords (“בית” is “house” and “הבית” is “the house”, so the first character is a prefix of a lexical unit). In addition, there are specific endings for gendered forms, which are highly regular and can be treated automatically. Therefore, the jamcode coding routine for Arabic and Hebrew defines a range of common prefixes and suffixes in each language, which are automatically considered (i.e., the keyword “בית” automatically matches also “הבית” or “ביתך” even if no truncation is defined). Specifically, the following affixes are pre-defined as part of the coding routine:

Arabic Prefixes	بال 'فال 'لل ال ل ن ي ت ل و ب ك م ف
Arabic Suffixes	ا ون نا تن تم ت ن وا ي كم ك ه هم ها كن ة ية ين ان
Hebrew Prefixes	מה נ ת י א ש ה מ ל כ ב ו
Hebrew Suffixes	ית ם ן ך ת תן תם תי ה כן כם הן הם נו ו כ י ות ים

Under normal circumstances, automatically permitting these prefixes and suffixes increases coding accuracy; however, every now and then, specific affixes create ambiguities, notably, if a word plus a specific prefix or suffix forms a homograph with another word that also exists.[3]

In such cases, the affix criteria serve to expressly *exclude* specific affixes:

_p(…)	This type of criterion excludes a specific prefix, so the keyword will not be matched if the character sequence is preceded by any listed character
_s(…)	This type of criterion excludes a specific suffix, so the keyword will not be matched if the character sequence is followed by any listed character

For instance, _p(מ) is satisfied if the final character before a recognized keyword is either a space or any of the listed prefixes, but *not* “מ”. Likewise, _s(ه) is satisfied if the first character following the recognized keyword is either a space or any of the listed suffixes, but *not* “ه”.

IV. Dictionary files

A dictionary can contain any number of concepts.

Each line defines a separate concept.

Every line follows a strict format of exactly four fields:

1. *Concept identifier*
   The concept identifier is a number that is unique to this concept, and will be written into most output formats, for efficient data handling.

2. *Concept label*
   The concept label can contain any free text, but neither tabs or line breaks. This label will be used as column head in the extended term document matrix format and in the annotated output format.

3. *Date criterion (optional)*
   Every concept can be restricted in time, i.e., the concept will only be coded in documents dated within a specified date range. Temporal criteria are specified as dd/mm/yy-dd/mm/yy, using double digits for single-digit days, months and years; years are assumed to be in the 2000s, i.e., 03/07/14 refers to 3 July 2014. Start date and end date can be identical (so only one day will be considered), but the end date cannot be before the start date. If the date criterion is empty, the concept will be coded regardless of the date.

   Please note:
   Unlike the disambiguation criterion (see above), this field governs the coding of the entire concept, and will be applied to every search phrase listed thereunder.

4. *Search phrases*
   This field contains all search phrases as defined above. There has to be at least one keyword, but there is no restriction to the number of keywords or keywords-plus-disambiguation-criteria search phrases permitted. All unique search phrases are separated by spaces. There is no particular order to these search phrases. A concept will be coded if the text matches any included search phrases (i.e., if the same word in a document matches multiple search phrases, it will still be coded only once).

   Please note:
   *To prevent redundant coding of adjacent words due to multiple matched search phrases (e.g., "not necessary" matches both "not_y(needed|necessary|important~5)" (the code will be applied to the keyword, "not"), and "necess*_y(no|none|neither|nor|never~10)" (the code will be applied to the keyword, "necess*"), the* jamcode *script includes a restriction that the same concept code will not be applied again within a word distance of 5 from an initial recognized position: If multiple search phrases identify the same concept in a multiword expression, the first word that matches a search phrase will be marked, and the subsequent ones will be ignored). This exclusion is lifted if another concept is coded in another word (i.e., if the first word is coded as concept A, the second is coded as concept B, then the third can again be coded as concept A). This exclusion is useful if the dictionary contains many multi-word expressions, and there are many coded concepts, so distinct nearby references to the same concept are usually separated by other coded concepts. For applications where this is not the case, the easiest way to circumvent this restriction is to add a final concept to the dictionary with * as the search phrase, which thus matches any non-space, non-punctuation character sequence in the document; this ensures that also nearby mentions of the same concept will always be separated (by this added concept), and thus correctly annotated.*
   Make sure that there are no unnecessary spaces in the search phrases field, i.e., neither double spaces, nor initial spaces, nor final spaces, nor any spaces within a search phrase.

The dictionary is passed to the jamcode coding script as a plain text file, named "DICT_<name>_<language>.txt", where <name> is a label for the dictionary, which should only contain letters or numbers, but no spaces, and <language> is a two-character, upper-case indicator of the dictionary's language (in case of the INFOCORE

Dictionary: AL (Albanian), AR (Arabic), DE (German), EN (English), FR (French), HE (Hebrew), MA (Macedonian), SR (Serbian))

Specifying language is important for two reasons.

1) If the language is recognized by the jamcode script, it will apply several language-sensitive harmonization scripts, which serve to efficiently handle special characters and diacritics. For instance, for Serbian, which uses either the Roman alphabet or a variant of the Cyrillic one, all text will be converted into a common character space; the jamcode script includes routines for handling language-specific ways of expressing acronums (e.g., the interspersed " in Hebrew), and other cleanup tools. If the language is not recognized by the script, it will be treated like English, i.e., with minimal intervention.
2) Identifying the language as Arabic (AR) or Hebrew (HE) switches on the pre-defined prefixes and affixes in the coding process (see above).

In the dictionary text file, every line defines one coded concept. There is no header line. Fields are separated by tabs. A dictionary has to contain at least one line. There cannot be any empty lines.

For instance, a minimal dictionary could look as follows (note the empty third, date criterion field):

```
1       Concept A               abc def_y(gh~2)
2       Concept B               ij kl_n(mn~5)_t(10/07/02-17/07/02)
```

V. Using the Dictionary

The dictionary is designed for use in conjunction with the jamcode python script available on Github (https://github.com/christianbaden/jamcode/). The jamcode coding script is designed for a located use of dictionaries: Beyond more conventional coding scripts, it determines not only whether a document contains a concept, but determines all instances of each concept, such that for every concept, its exact location in the document can be recorded.

Specifically, jamcode.py contains the main functions for loading the dictionary (importdict()), obtaining documents from a repository (gettexts()), preprocessing and tokenization (towords()), applying the dictionary (jcode()), and generating different kinds of output files (the default outputs are a results file that lists, for each document, which concept has been identified in which word position; and a term-document matrix that counts concept occurrences per document; in addition, e_annotate() generates text documents with annotations attached to all recognized concepts; e_replace() generates a file wherein each document is represented by the sequence of recognized concept identifiers; e_kwic() generates keywords in context for specified concepts). Drawing upon these functions, jcode.py is the governing script that conducts the analysis.

By default, jamcode interfaces with an AmCAT server, using the AmcatClient API. To import documents from other sources, the gettexts() function needs to be adjusted. Please refer to the jamcode Github site for additional documentation.