Understanding Enthymemes in Deductive Argumentation Using Semantic Distance Measures

Anthony Hunter

Department of Computer Science, University College London, London, WC1E 6BT, UK
anthony.hunter@ucl.ac.uk

Abstract

An argument can be regarded as some premises and a claim following from those premises. Normally, arguments exchanged by human agents are enthymemes, which generally means that some premises are implicit. So when an enthymeme is presented, the presenter expects that the recipient can identify the missing premises. An important kind of implicitness arises when a presenter assumes that two symbols denote the same, or nearly the same, concept (e.g. `dad` and `father`), and uses the symbols interchangeably. To model this process, we propose the use of semantic distance measures (e.g. based on a vector representation of word embeddings or a semantic network representation of words) to determine whether one symbol can be substituted by another. We present a theoretical framework for using substitutions, together with abduction of default knowledge, for understanding enthymemes based on deductive argumentation, and investigate how this could be used in practice.

Introduction

There are a number of frameworks for modelling argumentation in logic. They incorporate a formal representation of individual arguments, where the premises imply the claim, and techniques for comparing conflicting arguments (Atkinson et al. 2017). However, real arguments (i.e. arguments presented by humans) usually do not have enough explicitly presented premises for the entailment of the claim. This is because there is some common or commonsense knowledge that can be assumed by a proponent of an argument and the recipient of it (Walton 2001). This allows the proponent of an argument to encode it as an enthymeme by ignoring some of the common or commonsense knowledge, and it allows a recipient of the enthymeme to decode it into an argument by drawing on commonsense reasoning.

Whilst human agents constantly need to understand enthymemes, there is a lack of adequate AI methods for supporting or automating this process, The coding/decoding can be modelled as abduction (Hunter 2007; Black and Hunter 2012; Hosseini, Modgil, and Rodrigues 2014). However, these proposals assume an extensive set of common or commonsense knowledge with a preference ordering over the appropriateness of any formula for a specific audience. These are demanding assumptions. It is difficult to acquire commonsense knowledge, and in particular, there are often gaps in the explicit knowledge that connects concepts. Relationships between concepts are often known but they are implicit. For instance, different symbols for the same concept are used (e.g. `dad` and `father`) or for similar concepts (`showery` and `rainy`), though sometimes such correspondences are context dependent (e.g. in the context of a church, `dad` and `father` often refer to different concepts).

To address this need for a scalable and robust way of connecting concepts, we propose in this paper a solution based on distance measures between semantic concepts. We assume that each semantic concept is represented by a word, and so in propositional logic, each propositional atom is a word (or compound word), and in predicate logic, each predicate, function, and constant, symbol is a word (or compound word). So with a distance measure, we want the distance between `dad` and `father` to be low, and between `dad` and `grass` to be high. Fortunately, there are now some robust and scalable options for distance measures that we can use (as itemized below). None of these will be totally correct, but the error rate could be tolerable, and the benefits of using them far outweighing this.

Word embeddings Recent developments in deep learning allow learning of meanings of words in the form of word embeddings (e.g. Word2Vec (Mikolov et al. 2013) and GloVe (Pennington, Socher, and Manning 2014)) where the meaning of an individual word is represented by a vector. Each entry in the vector represents an aspect of the meaning of the word, though normally not in human meaningful terms. Rather it reflects co-occurrence information from the training data. Words can be compared by the cosine similarity of their vector representation (which is a distance measure), and so the smaller the value, the more similar the words are (according to the training data). Some word-embedding models are pre-trained on a large corpora of text, and have been shown to have good performance in recognizing similar concepts.

Semantic networks Whilst semantic networks have been considered in AI for decades, there has been a resurgence of interest in them (e.g. knowledge graphs (Paulheim 2017)) through the use of machine learning to construct them (e.g. the use of distantly supervised learning (Mintz et al. 2009)). A semantic network is a directed graph where each node denotes a concept or an attribute and each arc denotes a relationship between a pair of nodes. They can be constructed for specific applications or large general purpose networks can be harnessed such as WordNet which was developed for organizing over 100K words into sets of synonyms (called synsets) according to context plus hypernym and hyponym relationships (Miller 1995), or DBpedia which was developed to capture a wide range of concepts from Wikipedia including people, places, and organizations (Mendes et al. 2011). There are distance measures based on the structure of the graph (Budanitsky and Hirst 2006).

Logical ontologies Description logics offer more sophisticated options than semantic networks for representing and reasoning with knowledge about concepts. Some large ontologies have been developed that can be used as description logic ontologies such as SNOMED which has over 350K concepts in biomedicine (Rector and Brandt 2008). Various options for distance measures can be used based on the structure of the graph some of which are efficient to compute (Hu et al. 2006).

Lexical analysis There are a number of distance measures for comparing pairs of words (e.g. Hamming, Jaro, Levenshtein, or longest common sequence distance), and these offer cheap and scalable options for distance measures (e.g. (Petroni 2010)). For some applications (e.g. place names), it may be appropriate to use such distance measures for determining whether two words are sufficiently similar for them to represent the same concept.

The choice of distance measure depends on how we organize the words. There are advantages and disadvantages with each - in particular with regard to scalability (i.e. the scale of the problems they can be used for), reliability (i.e. the proportion of words that are correctly identified as denoting very similar concepts), and applicability (i.e. they are usable in a wide range of domains or only focused domains). What we can see is that there are options for identifying similar words from sets of 100K or more words, across broad domains, with reasonable reliability.

In the rest of the paper, we will review deductive argumentation, and then introduce our solution to understanding enthymemes in two parts. First, we will investigate how we can use semantic distance measures to find connections between logical symbols, and thereby allow for connected symbols to be substituted and so draw inferences that could not otherwise be made, and second, we will use this with abduction to find missing premises for enthymemes.

Preliminaries

We assume a finite set of function symbols $\mathcal{F}$. If the arity of a function symbol is zero, then it is a constant symbol. A term is a constant symbol or it is of the form $f(t_1,\ldots,t_n)$ where $f \in \mathcal{F}$ and $t_1,\ldots,t_n$ are terms. We also assume a finite set of predicate symbols $\mathcal{P}$. A predicate symbol has arity greater than or equal to zero. A zero arity predicate symbol is a propositional atom (e.g. `bird` or `fly`). A positive literal is of the form $p(t_1,\ldots,t_n)$ where $p \in \mathcal{P}$ and $t_1,\ldots,t_n$ are terms. Since we have no variables, each positive literal is ground, and it can be treated as a propositional atom. (e.g. $\texttt{like}(\texttt{Harry},\texttt{Sally})$). Let $\mathcal{A}$ be the set of atoms (i.e. propositional atoms and positive ground literals) that can be formed from $\mathcal{F}$ and $\mathcal{P}$.

Let $\mathcal{L}$ be the set of propositional formulae composed from atoms $\mathcal{A}$ and the logical connectives $\wedge$, $\vee$, $\neg$. We use $\alpha,\beta,\gamma,\delta,\phi,\psi,\ldots$ for arbitrary formulae and $\Delta,\Phi,\Psi,\ldots$ for arbitrary sets of formulae. A knowledgebase $\Delta \subseteq \mathcal{L}$ is a finite set of formulae. We use $\mathcal{K}$ for the set of all knowledgebases. We let $\vdash$ denote the classical consequence relation, and write $\Delta \vdash \bot$ to denote that Δ is inconsistent. $\mathsf{Atoms}(\Delta)$ gives the atoms appearing in the formulae in Δ. Let Cn be the consequence closure function (i.e. $\mathsf{Cn}(\Delta) = \{\phi \mid \Delta \vdash \phi\}$). For $\phi,\psi \in \mathcal{L}$, $\phi \equiv \psi$ denotes that ϕ and ψ are equivalent (i.e. $\{\phi\} \vdash \psi$ and $\{\psi\} \vdash \phi$). For $\Phi,\Psi \subseteq \mathcal{L}$, $\Phi \equiv \Psi$ denotes that Φ and Ψ are equivalent (i.e. $\mathsf{Cn}(\Phi) = \mathsf{Cn}(\Psi)$).

Deductive Argumentation

In deductive argumentation, an **argument** is a pair $\langle\Phi,\alpha\rangle$ where $\Phi \subseteq \mathcal{L}$ is a minimal set such that Φ is consistent and Φ entails the claim α (Besnard and Hunter 2001). In this paper, we adapt a version of deductive argumentation that was proposed for handling enthymemes (Hunter 2007). An **approximate argument** is a pair $\langle\Phi,\alpha\rangle$ where $\Phi \subseteq \mathcal{L}$ and $\alpha \in \mathcal{L}$. This is a very general definition. It does not assume that Φ is consistent, or that it even entails α. An argument is a special case of an approximate argument. For an approximate argument $\langle\Phi,\alpha\rangle$, let $\mathsf{Support}(\langle\Phi,\alpha\rangle)$ be Φ, and let $\mathsf{Claim}(\langle\Phi,\alpha\rangle)$ be α.

Some kinds of approximate arguments include: If $\Phi \vdash \alpha$, then $\langle\Phi,\alpha\rangle$ is **valid**; If $\Phi \nvdash \bot$, then $\langle\Phi,\alpha\rangle$ is **consistent**; If $\Phi \vdash \alpha$, and there is no $\Phi' \subset \Phi$ such that $\Phi' \vdash \alpha$, then $\langle\Phi,\alpha\rangle$ is **minimal**; And if $\Phi \vdash \alpha$, and $\Phi \nvdash \bot$, then $\langle\Phi,\alpha\rangle$ is **expansive** (i.e. it is valid and consistent, but it may have unnecessary premises).

In addition, we require a further kind of approximate argument that has the potential to be transformed into an argument: If $\Phi \nvdash \alpha$, and $\Phi \nvdash \neg\alpha$, then $\langle\Phi,\alpha\rangle$ is a **precursor** (i.e. it is a precursor for an argument). Therefore, if $\langle\Phi,\alpha\rangle$ is a precursor, then there exists some $\Psi \subset \mathcal{L}$ such that $\Phi \cup \Psi \vdash \alpha$ and $\Phi \cup \Psi \nvdash \bot$, and hence $\langle\Phi \cup \Psi,\alpha\rangle$ is expansive.

Example 1. *Let* $\Delta = \{\mathsf{a}, \neg\mathsf{a} \vee \mathsf{b}, \mathsf{c}, \neg\mathsf{b}, \mathsf{b}, \neg\mathsf{c}, \neg\mathsf{b} \vee \mathsf{c}\}$. *Some approximate arguments from* Δ *that are valid include* $\{A_1, A_2, A_3, A_4, A_5, A_8\}$ *of which* $\{A_1, A_3, A_4, A_8\}$ *are expansive,* $\{A_2, A_4, A_8\}$ *are minimal, and* $\{A_4, A_8\}$ *are arguments. Also, some approximate arguments that are not valid include* $\{A_6, A_7\}$ *of which* A_6 *is a precursor.*

$A_1 = \langle\{\mathsf{a}, \neg\mathsf{a} \vee \mathsf{b}, \mathsf{c}, \mathsf{b}\}, \mathsf{b}\rangle$ $\quad A_2 = \langle\{\mathsf{c}, \neg\mathsf{c}\}, \mathsf{b}\rangle$
$A_3 = \langle\{\mathsf{a}, \neg\mathsf{a} \vee \mathsf{b}, \mathsf{c}\}, \mathsf{b}\rangle$ $\quad A_4 = \langle\{\mathsf{a}, \neg\mathsf{a} \vee \mathsf{b}\}, \mathsf{b}\rangle$
$A_5 = \langle\{\mathsf{a}, \neg\mathsf{a} \vee \mathsf{b}, \mathsf{c}, \neg\mathsf{c}\}, \mathsf{b}\rangle$ $\quad A_6 = \langle\{\neg\mathsf{a} \vee \mathsf{b}\}, \mathsf{b}\rangle$
$A_7 = \langle\{\neg\mathsf{a} \vee \mathsf{b}, \neg\mathsf{b} \vee \mathsf{c}, \neg\mathsf{c}\}, \mathsf{b}\rangle$ $\quad A_8 = \langle\{\}, \mathsf{a} \vee \neg\mathsf{a}\rangle$

Some observations concerning approximate arguments include: (1) If $\langle\Gamma,\alpha\rangle$ is expansive, then there is a $\Phi \subseteq \Gamma$ such that $\langle\Phi,\alpha\rangle$ is an argument; (2) If $\langle\Phi,\alpha\rangle$ is minimal, and $\langle\Phi,\alpha\rangle$ is expansive, then $\langle\Phi,\alpha\rangle$ is an argument; (3) If $\langle\Phi,\alpha\rangle$ is an argument, and $\Psi \subset \Phi$, then $\langle\Psi,\alpha\rangle$ is a precursor; and (4) If $\langle\Gamma,\alpha\rangle$ is a precursor, then $\langle\Gamma,\alpha\rangle$ is consistent.

An **enthymeme** is a precursor that can be generated from an argument (i.e. a precursor $\langle\Phi,\alpha\rangle$ is an enthymeme for argument $\langle\Psi,\alpha\rangle$ iff $\Phi\subset\Psi$). So if a proponent has an argument that it wishes to present to a recipient, the **intended argument**, then the proponent may send an enthymeme instead of the intended argument to the recipient.

Example 2. *Let* $\mathtt{u}$ *be "you need an umbrella today", and* $\mathtt{r}$ *be "the weather report predicts rain". So for an intended argument* $\langle\{\mathtt{r},\mathtt{r}\rightarrow\mathtt{u}\},\mathtt{u}\rangle$*, the enthymeme sent by the proponent to the recipient may be* $\langle\{\mathtt{r}\},\mathtt{u}\rangle$

Since there can be more than one enthymeme that can be generated from an intended argument, a proponent needs to choose which to send to a recipient. To facilitate this selection, the proponent consults what it believes to be common or commonsense knowledge. We assume that each agent has a personal knowledgebase Δ, and a common/commonsense knowledgebase Π. Now consider an agent with intended argument $\langle\Phi,\alpha\rangle$ that it wants to send to a recipient. So Φ is a subset of Δ. The agent will remove premises from Φ that are common/commonsense knowledge. So for an argument $\langle\Phi,\alpha\rangle$, the **encodation** of $\langle\Phi,\alpha\rangle$ from a proponent is the approximate argument $\langle\Psi,\alpha\rangle$, where Ψ is $\Phi\setminus\Pi$. The result of this encodation process is either the intended argument or an enthymeme for that argument. In Example 2, the encoded argument is $\langle\{\mathtt{r}\},\mathtt{u}\rangle$ when $\mathtt{r}\rightarrow\mathtt{u}\in\Pi$.

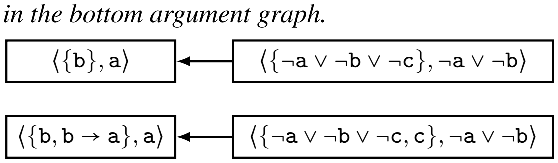
Next, we review some types of **counterargument** taken from (Gorogiannis and Hunter 2011). where A and B are approximate arguments: A is a **defeater** of B if $\mathsf{Claim}(A)\vdash\neg\bigwedge\mathsf{Support}(B)$; A is an **undercut** of B if $\exists\Psi\subseteq\mathsf{Support}(B)$ s.t. $\mathsf{Claim}(A)\equiv\neg\bigwedge\Psi$; A is a **direct undercut** of B if $\exists\phi\in\mathsf{Support}(B)$ s.t. $\mathsf{Claim}(A)\equiv\neg\phi$; A is a **defeating rebuttal** of B if $\mathsf{Claim}(A)\vdash\neg\mathsf{Claim}(B)$.

Example 3. *Let* $\Delta=\{\mathsf{a}\vee\mathsf{b},\mathsf{a}\leftrightarrow\mathsf{b},\mathsf{c}\rightarrow\mathsf{a},\neg\mathsf{a}\wedge\neg\mathsf{b},\mathsf{a},\mathsf{b},\mathsf{c},\mathsf{a}\rightarrow\mathsf{b},\neg\mathsf{a},\neg\mathsf{b},\neg\mathsf{c}\}$. $\langle\{\mathsf{a}\vee\mathsf{b},\mathsf{c}\},(\mathsf{a}\vee\mathsf{b})\wedge\mathsf{c}\rangle$ *is a defeater of* $\langle\{\neg\mathsf{a},\neg\mathsf{b}\},\neg\mathsf{a}\wedge\neg\mathsf{b}\rangle$*,* $\langle\{\neg\mathsf{a}\wedge\neg\mathsf{b}\},\neg(\mathsf{a}\wedge\mathsf{b})\rangle$ *is a undercut of* $\langle\{\mathsf{a},\mathsf{b},\mathsf{c}\},\mathsf{a}\wedge\mathsf{b}\wedge\mathsf{c}\rangle$*,* $\langle\{\neg\mathsf{a}\wedge\neg\mathsf{b}\},\neg\mathsf{a}\rangle$ *is a direct undercut of* $\langle\{\mathsf{a},\mathsf{b},\mathsf{c}\},\mathsf{a}\wedge\mathsf{b}\wedge\mathsf{c}\rangle$*, and* $\langle\{\mathsf{a},\mathsf{a}\rightarrow\mathsf{b}\},\mathsf{b}\rangle$ *is a defeating rebuttal of* $\langle\{\neg\mathsf{a}\wedge\neg\mathsf{b},\neg\mathsf{c}\},\neg(\mathsf{b}\vee\mathsf{c})\rangle$*.*

In this paper, an **argument graph** is a graph where each node is an approximate argument, and each arc denotes an attack as identified on the intended argument. Extensions can be identified using Dung's semantics (Dung 1995).

We assume that some or all of the approximate arguments in the argument graph are given to an agent (the recipient). For instance, the graph could be constructed incrementally in a dialogue: With each cycle of the dialogue, an agent could receive an enthymeme from the other agent, which it needs to understand, and then it responds with its own approximate argument.

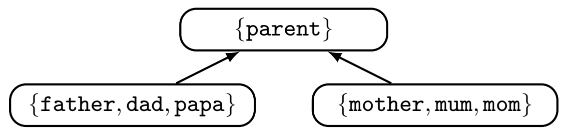
Example 4. *Consider an agent receiving the enthymeme at the root in the top argument graph, and replying with the enthymeme at the leaf. The corresponding intended arguments are in the bottom argument graph.*



Understanding via Substitution

For a set of predicate symbols $\mathcal{P}$ and a set of function symbols $\mathcal{F}$, a **distance measure** D is a function $D:\mathcal{P}\cup\mathcal{F}\times\mathcal{P}\cup\mathcal{F}\rightarrow[0,\infty]$ that satisfies: (identity of indiscernibles) $D(s_1,s_2)=0$ when $s_1=s_2$; (symmetry) $D(s_1,s_2)=D(s_2,s_1)$; and (triangle inequality) $D(s_1,s_2)\leq D(s_1,s_3)+D(s_3,s_2)$. We assume that the predicate and function symbols are words that represent concepts, and we use a distance measure to capture semantic distance between these concepts. The nearer two symbols (i.e. words) are according to the distance measure, the closer the concepts conveyed by the symbols.

Example 5. *In the following semantic network, each node is a set of synonyms, and an arc denotes the hyponym relationship. The distance measure D is the number of arcs traversed in connecting two words. Hence,* $D(\mathtt{dad},\mathtt{father})=0$*,* $D(\mathtt{dad},\mathtt{parent})=1$*, and* $D(\mathtt{dad},\mathtt{mum})=2$*.*



We use a distance measure to specify changes in the symbols in formulae. A **swap** is a pair of symbols, denoted s_1/s_2 where $s_1,s_2\in\mathcal{P}$, or $s_1,s_2\in\mathcal{F}$. For each swap s_1/s_2, we call s_1 the **outgoing symbol** and s_2 the **incoming symbol**. When a swap s_1/s_2 is applied to a formula ϕ, s_1 is substituted by s_2 throughout ϕ. So for example, for $\mathtt{father(John,Sue)}$, the application of $\mathtt{father/dad}$ will result in $\mathtt{dad(John,Sue)}$, and the application of $\mathtt{Sue/Susan}$ will result in $\mathtt{dad(John,Susan)}$. A swap s_1/s_2 is **reflexive** iff $s_1=s_2$. A swap s_1/s_2 is a **valid swap** for distance measure D and threshold $\tau\in[0,\infty]$ iff $D(s_1,s_2)\leq\tau$. Hence, for a swap s_1/s_2, if $s_1=s_2$ or $\tau=\infty$, then s_1/s_2 is valid.

A **substitution** is a set of swaps $\{s_1/s_1',\ldots,s_n/s_n'\}$ such that there is at most one swap for each outgoing symbol (i.e. for all $s_i/s_i',s_j/s_j'\in\{s_1/s_1',\ldots,s_n/s_n'\}$, $s_i\neq s_j$), and no symbol is both incoming and outgoing unless it is for a reflexive swap (i.e. for all $s_i/s_i',s_j/s_j'\in\{s_1/s_1',\ldots,s_n/s_n'\}$, $s_i'\neq s_j$ unless s_i/s_i' and s_j/s_j' are reflexive). For example, $\{\mathsf{a}/\mathsf{b},\mathsf{d}/\mathsf{c}\}$ is a substitution, whereas $\{\mathsf{a}/\mathsf{b},\mathsf{a}/\mathsf{c}\}$ and $\{\mathsf{a}/\mathsf{b},\mathsf{b}/\mathsf{c}\}$ are not. Let $\mathcal{S}$ be the set of all substitutions. A substitution $\{s_1/s_1',\ldots,s_n/s_n'\}$ is a **valid substitution** for distance measure D and threshold $\tau\in[0,\infty]$ iff each $s_i/s_i'\in\{s_1/s_1',\ldots,s_n/s_n'\}$ is valid for D and τ.

The next definition is a set of rewrite rules for applying a substitution to an arbitrary formula and to sets of formulae. If the same symbol occurs multiple times, then all occurrences of that symbol are changed in the same way.

Definition 1. *Let S be a valid substitution. The* **substitute operator***, denoted $\oplus$, is defined by rewrite rules as follows: (1) for a term $f(t_1,\ldots,t_n)$ where f is arity zero or more, $f(t_1,\ldots,t_n)\oplus S\Rightarrow f'(t_1\oplus S,\ldots,t_n\oplus S)$ where $f/f'\in S$ otherwise $f=f'$; (2) for an atom $p(t_1,\ldots,t_n)$ where p is arity zero or more, $p(t_1,\ldots,t_n)\oplus S\Rightarrow p'(t_1\oplus S,\ldots,t_n\oplus S)$ where $p/p'\in S$ otherwise $p=p'$; (3) for a formula $\neg\phi$, $\neg\phi\oplus S\Rightarrow\neg(\phi\oplus S)$; (4) for formulae $\phi\#\psi$, where $\#\in\{\wedge,\vee,\rightarrow\}$,*

$\phi \# \psi \oplus S \Rightarrow (\phi \oplus S) \# (\psi \oplus S)$*; and (5) for a set of formulae* Γ, $\Gamma \oplus S \Rightarrow \{\phi \oplus S' \mid \phi \in \Gamma$ *and* $S' \subseteq S\}$.

So the rewrite rules are applied exhaustively, and the result is the substituted formula (or set of formulae).

Example 6. *For* $S = \{\texttt{father/dad}, \texttt{Sue/Susan}\}$ *we get* $\texttt{father(John, Sue)} \rightarrow \texttt{parent(John, Sue)} \oplus S$ *rewritten to* $\texttt{dad(John, Susan)} \rightarrow \texttt{parent(John, Susan)}$ *and* $\texttt{father} \rightarrow \texttt{parent} \oplus S$ *rewritten to* $\texttt{dad} \rightarrow \texttt{parent}$.

Example 7. *For* $S = \{\texttt{driven/chauffeured}, \texttt{John/Jon}\}$ *we get* $\texttt{driven(the\_car(red), by(John))} \oplus S$ *is rewritten to* $\texttt{chauffeured(the\_car(red), by(Jon))}$.

Example 8. *For* $\Gamma = \{\texttt{a(c, e)}\}$ *and* $S = \{\texttt{a/b}, \texttt{c/d}, \texttt{g/h}\}$, *we get* $\Gamma \oplus S = \{\texttt{a(c, e)}, \texttt{b(c, e)}, \texttt{a(d, e)}, \texttt{b(d, e)}\}$.

Substitution can be used on a lazy basis. In other words, if we have an enthymeme that we wish to understand, we look for appropriate substitutions in order to decode it. An alternative would be to augment a knowledgebase with all variants of the formula in the knowledgebase. For example, if we have the formula $\texttt{dad} \rightarrow \texttt{parent}$ in the knowledgebase, and the swap $\texttt{dad/father}$, then we could add variants such as $\texttt{father} \rightarrow \texttt{parent}$ to the knowledgebase.

However, augmenting the knowledgebase may involve adding a very large number of variants, and thereby make it difficult to maintain and use. For instance, suppose the knowledgebase is a set of k clauses each with 4 disjuncts (i.e. formulae of the form $\phi_1 \vee \phi_2 \vee \phi_3 \vee \phi_4$), and each disjunct ϕ_i has n valid swaps, then the knowledgebase would have $k \times n^4$ variants. So if there are 100 clauses in the knowledgebase, and each disjunct has 4 synonyms, then the augmented knowledgebase would have $100 \times 5^4 = 62.5$K variants. So there is a distinct advantage in using subsitution as required rather than augmenting the knowledgebase.

We now consider properties of substitution to show that the operation is well-behaved. The empty substitution is idempotent (i.e. if $S = \varnothing$, then $\phi \oplus S = \phi$). An identity substituion is also idempotent (i.e. if $S = \{s_1/s_1, \ldots, s_n/s_n\}$, then $\phi \oplus S = \phi$). Since S is based on the symbols of the language, the language is closed under substitutions (i.e. $\mathcal{L} = \mathcal{L} \oplus S$ for all substitutions $S \in \mathcal{S}$). However, substitution is not reversible in general: For formula ϕ and substitution S, there is not necessarily a substitution S' such that $(\phi \oplus S) \oplus S' = \phi$ as illustrated by the following example.

Example 9. *Let* $\phi = \texttt{nickname(Andrew, Andy)}$ *with the substitution* $S = \texttt{Andy/Andrew}$ *means* $\phi \oplus S = \texttt{nickname(Andrew, Andrew)}$. *So there is no substitution* S' *such that* $(\phi \oplus S) \oplus S' = \phi$.

Substitution does not decrease the consequences from the knowledge.

Proposition 1. *For* $\Gamma \in \mathcal{K}$, $S \in \mathcal{S}$, $\mathsf{Cn}(\Gamma) \subseteq \mathsf{Cn}(\Gamma \oplus S)$.

Proof. By definition, $\Gamma \oplus S \Rightarrow \{\phi \oplus S' \mid \phi \in \Gamma$ and $S' \subseteq S\}$. Also $\Gamma = \{\phi \oplus \varnothing \mid \phi \in \Gamma\}$. Hence, $\mathsf{Cn}(\Gamma) \subseteq \mathsf{Cn}(\Gamma \oplus S)$. □

As a corollary, if A is an argument from Γ (i.e. $\mathsf{Support}(A) \subseteq \Gamma$), then A is an argument from $\Gamma \oplus S$ for any substitution S. However, a substitution can cause a consistent set of formulae to become inconsistent as in the following example.

Example 10. *For* $S = \{\texttt{Nik/Niki}, \texttt{dad/father}\}$, *and* $\Gamma = \{\texttt{dad(Nik, Jo)}, \neg\texttt{father(Niki, Jo)}\}$, $\Gamma \oplus S$ *is* $\{\texttt{dad(Nik, Jo)}, \neg\texttt{father(Niki, Jo)}, \texttt{father(Nik, Jo)}, \texttt{dad(Niki, Jo)}, \texttt{father(Niki, Jo)}\}$ *which is inconsistent.*

Consequences are maintained under substitution as shown next, and hence there is equivalence over formulae under substitution (i.e. if $\phi \equiv \psi$, then $\phi \oplus S \equiv \psi \oplus S$).

Proposition 2. *Let* $S \in \mathcal{S}$. *If* $\Gamma \vdash \phi$, *then* $\Gamma \oplus S \vdash \phi \oplus S$

Proof. For each set of formulae Γ, there is a set of clauses Ψ such that $\Gamma \equiv \Psi$. So $\Gamma \vdash \phi$ iff $\Psi \vdash \phi$. For clauses $\sigma \vee \phi_1 \vee \ldots \vee \phi_n$ and $\neg\sigma \vee \psi_1 \vee \ldots \vee \psi_m$, $\phi_1 \vee \ldots \vee \phi_n \vee \psi_1 \vee \ldots \vee \psi_m$ is a resolvent. Let $\mathsf{Resolvents}(\Psi)$ be the set of resolvents of Ψ. So $\Psi \vdash \phi$ iff $\mathsf{Resolvents}(\Psi \cup \Theta) \vdash \bot$ where Θ is the set of clauses obtained from the negation of ϕ (i.e. $\Theta \equiv \{\neg\phi\}$). Furthermore, if $\psi \in \mathsf{Resolvents}(\Psi)$, then $\psi \oplus S \in \mathsf{Resolvents}(\Psi \oplus S)$. So if $\mathsf{Resolvents}(\Psi \cup \Theta) \vdash \bot$, then $\mathsf{Resolvents}((\Psi \cup \Theta) \oplus S) \vdash \bot$. Hence, $\Gamma \vdash \phi$ implies $\Gamma \oplus S \vdash \phi \oplus S$. □

Proposition 3. *Let* $S \in \mathcal{S}$. $\mathsf{Cn}(\Gamma) \oplus S \subseteq \mathsf{Cn}(\Gamma \oplus S)$.

Proof. We extend the proof for Proposition 2. Assume $\psi \in \mathsf{Cn}(\Gamma)$. So if $\Theta \equiv \{\neg\psi\}$ and $\mathsf{Resolvents}(\Psi \cup \Theta) \vdash \bot$, then $\mathsf{Resolvents}((\Psi \cup \Theta) \oplus S) \vdash \bot$. So $\psi \oplus S \in \mathsf{Cn}(\Gamma \oplus S)$. Therefore $\mathsf{Cn}(\Gamma) \oplus S \subseteq \mathsf{Cn}(\Gamma \oplus S)$. Now, we consider a counterexample for $\mathsf{Cn}(\Gamma \oplus S) \subseteq \mathsf{Cn}(\Gamma) \oplus S$. Let $\Gamma = \{\texttt{a} \vee \texttt{b}, \neg\texttt{c} \vee \texttt{d}\}$ and $S = \{\texttt{c/b}\}$. So $\{\texttt{a} \vee \texttt{d}\} \in \mathsf{Cn}(\Gamma \oplus \texttt{S})$ but $\{\texttt{a} \vee \texttt{d}\} \notin \mathsf{Cn}(\Gamma) \oplus \texttt{S}$. □

Substitution is monotonic. So increasing the number of swaps does not decrease the set of inferences. This is a generalization of Proposition 2.

Proposition 4. *Let* $S, S' \in \mathcal{S}$. *If* $\Gamma \oplus S \vdash \phi \oplus S$, *and* $S \subseteq S'$, *then* $\Gamma \oplus S' \vdash \phi \oplus S$.

However, if the set of outgoing symbols is maintained, but the set of incoming symbols is decreased, then the set of inferences does not necessarily increase or decrease.

Example 11. *Let* $\Gamma = \{\texttt{a}, \texttt{b} \rightarrow \texttt{c}, \texttt{d} \rightarrow \texttt{e}\}$. *For* $S_1 = \{\texttt{b/a}, \texttt{d/f}\}$ *and* $S_2 = \{\texttt{b/a}, \texttt{d/a}\}$, *the set of outgoing symbols is the same in both substitutions, but the set of incoming symbols is decreased in* S_2, *and* $\mathsf{Cn}(\Gamma \oplus S_1)$ *is neither a subset nor superset of* $\mathsf{Cn}(\Gamma \oplus S_2)$.

Counterargument relationships (i.e attack relationships) are also maintained under substitution as shown next.

Proposition 5. *Let* $S \in \mathcal{S}$. *If* $\langle\Phi, \alpha\rangle$ *is a defeater (resp. undercut, direct undercut, or defeating rebuttal) for* $\langle\Psi, \beta\rangle$, *then* $\langle\Phi \oplus S, \alpha \oplus S\rangle$ *is a defeater (resp. undercut, direct undercut, or defeating rebuttal) for* $\langle\Psi \oplus S, \beta \oplus S\rangle$.

Proof. Assume $\langle\Phi, \alpha\rangle$ is a defeater for $\langle\Psi, \beta\rangle$. Hence $\alpha \vdash \neg \bigwedge \Psi$. Therefore, by Proposition 2, $\alpha \oplus S \vdash (\neg \bigwedge \Psi) \oplus S$. Also by Proposition 2, $\Phi \vdash \alpha$ implies $\Phi \oplus S \vdash \alpha \oplus S$, and $\Psi \vdash \beta$ implies $\Psi \oplus S \vdash \beta \oplus S$. Therefore, $\langle\Phi \oplus S, \alpha \oplus S\rangle$ is a defeater for $\langle\Psi \oplus S, \beta \oplus S\rangle$. Proofs for undercut, direct undercut, and defeating rebuttal, are similar. □

When we assume that the set of symbols is finite, it is straightforward to show that substitution is decidable.

build	construct	0.26	fabricate	construct	0.7
father	dad	0.37	house	bungalow	0.72
house	home	0.5	train	school	0.72
like	love	0.51	saloon	barroom	0.73
fish	water	0.54	fish	apple	0.79
saloon	bar	0.61	teacher	tree	0.88
father	parent	0.69	truck	cat	0.91

Table 1: Examples using the pre-trained GloVe word embedding (6B.300d). Each entry is a pair of words and the cosine similarity distance between them.

Proposition 6. *Determining whether or not there is a substitution $S \subseteq \mathcal{S}$ such that $\Gamma \oplus S \vdash \phi \oplus S$ holds is decidable when $\Gamma \in \mathcal{K}$ and and $\phi \in \mathcal{L}$.*

Proof. Assume the set of atoms $\mathcal{A}$ is finite. So the set of substitutions $\mathcal{S}$ is finite. For each $S \in \mathcal{S}$, determining $\Gamma \oplus S \vdash \phi \oplus S$ is decidable for propositional logic. So determining whether or not there is a substitution S such that $\Gamma \oplus S \vdash \phi \oplus S$ holds is decidable. □

Word embeddings are a promising way of finding substitutions (see Table 1 for examples). Pre-trained embeddings contain vocabularies of 100K or more words, and are trained across wide varieties of text (e.g. GloVe (Pennington, Socher, and Manning 2014)). As seen in Table 1, we might set a threshold under 0.75 for getting useful swaps. Though swaps might not always be correct as they reflect words that appear frequently in the same text but are not closely related concepts (e.g. for `fish` and `water`) or they only work in some contexts (e.g. for `train` and `school`).

In order to investigate the viability of using pre-trained word embeddings, we can use WordNet to compare the distances obtained when pairs of words are in the same synset, and when they are in different synsets. So we use WordNet as the gold standard for pairs of synonyms and pairs of non-synonyms. Even when we restrict the number of arcs between the two different synsets, the average distance (cosine similarity of the word vector representation) between different words in the same synset are lower than different words in different synsets. As an example, for the subnetwork of WordNet rooted at the noun `artefact`, there are 7319 synsets, and in these there are 4830 different non-compound nouns (i.e. tokens without underscore). There are fewer words than synsets because each word can occur in multiple synsets. In this example, for pairs of different words in the same (respectively different) synset(s), the mean distance according to GloVe is 0.79 with standard deviation of 0.19 (respectively 0.97 with standard deviation of 0.10). Moveover, any pair of words with a distance of 0.8 (or below) is much more likely to be in the same synset than not. Similar results were obtained for other nouns in other parts of WordNet and for verbs (see appendix[1]). If GloVe or Word2Vec is trained for a specific domain, then a substantial improvement in discrimination between closely related concepts and less closely related concepts could be expected.

[1]Appendix: www0.cs.ucl.ac.uk/staff/a.hunter/papers/aaai22.zip

A key advantage of word embeddings is that they provide connections between words based on co-occurrence, and so provide a scaleable and quite robust proxy for when one word can be substituted for another. Furthermore, these are not restricted to identifying synonyms but also include identifying a wider range of closely related concepts. However, substitutions as presented in this section, are not qualified by context, and may make some incorrect connections.

Understanding via Abduction

The substitution approach involves changing formulae to allow inferences to be made. Essentially, it allows for connections between atoms arising in the formulae in the premises. However, in order to understand some enthymemes, we also need to bring in further default knowledge via abduction. Consider the enthymeme $\langle\{\texttt{eagle}\}, \texttt{fly}\rangle$. Suppose we have default knowledge that includes $\texttt{bird} \rightarrow \texttt{fly}$, and $D(\texttt{eagle}, \texttt{bird}) < \tau$ (and so $\texttt{eagle}/\texttt{bird}$ is a valid swap), then by abduction we can identify this knowledge.

Definition 2. *An* **abductive solution** *for an approximate argument $\langle\Phi, \psi\rangle$ is a pair (Γ, S) where $\Gamma \subseteq \Pi$ and Π is a set of default knowledge and a valid $S \subseteq \mathcal{S}$ w.r.t. distance measure D and threshold τ such that*

- $(\Phi \cup \Gamma) \oplus S \vdash \psi$ *(entailment)*
- $(\Phi \cup \Gamma') \oplus S \not\vdash \psi$ *for all $\Gamma' \subset \Gamma$ (min assumptions)*
- $(\Phi \cup \Gamma) \oplus S' \not\vdash \psi$ *for all $S' \subset S$ (min swaps)*

An abductive solution (Γ, S) is **consistent** *for $\langle\Phi, \psi\rangle$ iff $(\Phi \cup \Gamma) \oplus S \not\vdash \bot$.*

Obviously, there can be multiple abductive solutions for an enthymeme $\langle\Phi, \psi\rangle$. Also, it is possible for an abductive solution (Γ, S) to be such that either $\Gamma = \varnothing$ or $S = \varnothing$. However, (Γ, S) is an abductive solution for $\langle\Phi, \psi\rangle$ where $\Phi \not\vdash \psi$ iff $\Gamma \neq \varnothing$ or $S \neq \varnothing$. If $\langle\Phi, \psi\rangle$ is inconsistent, or if $\langle\Phi, \psi\rangle$ is valid, then there is always an abductive solution (Γ, S) for $\langle\Phi, \psi\rangle$ where $\Gamma = \varnothing$ and $S = \varnothing$. Also, for any argument $\langle\Phi, \psi\rangle$, if there is a formula ψ' such that $\Phi \vdash \psi'$, and ψ and ψ' have isomorphic syntax trees, then there is an abductive solution $(\varnothing, S)$ (i.e. $\psi \oplus S \equiv \psi'$).

We also may want to make the connection identified by the distance measure explicit. For this, we use the following definition to decode the abductive solution for an approximate argument. Essentially, for each atom ϕ appearing as subformulae in the formula in the premises and default knowledge, an implicational formula of the form $\phi \rightarrow (\phi \oplus S')$ is obtained where $S' \subseteq S$ and $S' \neq \varnothing$.

Definition 3. *For an abductive solution (Γ, S) for an approximate argument $\langle\Phi, \psi\rangle$, a* **decoded argument** *is $\langle\Phi \cup \Gamma \cup \mathsf{Ex}(\Phi, \Gamma, S), \psi\rangle$ where $\mathsf{Ex}(\Phi, \Gamma, S) = \{\phi \rightarrow (\phi \oplus S) \mid \phi \in \mathsf{Atoms}(\Phi \cup \Gamma)$ and $S' \subseteq S$ and $S' \neq \varnothing\}$.*

If each $\phi \in \mathsf{Atoms}(\Phi \cup \Gamma)$ is a propositional atom (i.e. not a positive ground literal), then $\mathsf{Ex}(\Phi, \Gamma, S) = \{\phi \rightarrow (\phi \oplus S) \mid \phi \in \mathsf{Atoms}(\Phi \cup \Gamma)\}$.

Example 12. *For enthymeme $\langle\{\texttt{eagle}\}, \texttt{fly}\rangle$, the decoded argument is $\langle\{\texttt{eagle}, \texttt{eagle} \rightarrow \texttt{bird}, \texttt{bird} \rightarrow \texttt{fly}\}, \texttt{fly}\rangle$. where $\texttt{bird} \rightarrow \texttt{fly}$ is default knowledge, and $D(\texttt{eagle}, \texttt{bird}) < \tau$ (i.e. $\texttt{eagle}/\texttt{bird} \in \mathcal{S}$).*

A decoded argument therefore incorporates knowledge about the substitution and default knowledge used in the abductive solution, and therefore provides the implicit knowledge that allows for the claim to be derived from the conclusions using the classical consequence relation as captured in the following result.

Proposition 7. *If* (Γ, S) *is an abductive solution for* $\langle \Phi, \psi \rangle$, *and* Δ, Θ, *and* Ψ *are sets of clauses s.t.* $\Delta \equiv \Phi \cup \Gamma$, $\Theta \equiv \{\neg\psi\}$, *and* $\Psi \equiv \mathsf{Ex}(\Phi, \Gamma, S)$, *then* $\mathsf{Resolvents}((\Delta \cup \Theta) \oplus S) = \mathsf{Resolvents}(\Delta \cup \Psi \cup \Theta)$.

Proof. $\phi \in \mathsf{Resolvents}((\Delta \cup \Theta) \oplus S)$, iff there are $\phi', \phi'' \in \mathsf{Resolvents}((\Delta \cup \Theta) \oplus S)$ s.t. ϕ is a resolvent of ϕ' and ϕ'', iff there is an $\Omega \subseteq (\Delta \cup \Theta) \oplus S$ s.t. $\phi \in \mathsf{Resolvents}(\Omega)$, iff there is a $\Xi \subseteq (\Delta \cup \Theta)$ and $S' \subseteq S$ s.t. $\phi \in \mathsf{Resolvents}(\Xi \oplus S')$, iff there is an $\Upsilon \subseteq (\Delta \cup \Theta)$ and $\Psi' \subseteq \Psi$ s.t. $\phi \in \mathsf{Resolvents}(\Upsilon \cup \Psi')$, iff $\phi \in \mathsf{Resolvents}(\Delta \cup \Psi \cup \Theta)$ □

The above result implies that entailment and consistency coincide for an argument and a decoded argument, and hence we get the following.

Proposition 8. *If* (Γ, S) *is an abductive solution (resp. a consistent abductive solution) for* $\langle \Phi, \psi \rangle$, *then* $\langle \Phi \cup \Gamma \cup \mathsf{Ex}(\Phi, \Gamma, S), \psi \rangle$ *is valid (resp. expansive).*

To gain insights into the nature of abductive solutions, we view abductive solutions via a consequence relation.

Definition 4. *Let* D *be a distance measure and* τ *be a threshold. The* **enthymeme inference relation***, denoted* $\Vdash^{\tau}_{D}$, *is defined as follows:* $\Phi \Vdash^{\tau}_{D} \psi$ *iff there is an abductive solution* (Γ, S) *for* $\langle \Phi, \psi \rangle$ *w.r.t* τ *and* D.

Obviously, the enthymeme inference relation collapses to the classical consequence relation when the abductive solution is empty (i.e. $\Phi \vdash \psi$ iff $\Gamma = \emptyset$ and $S = \emptyset$ and $\Phi \Vdash^{\tau}_{D} \psi$). Also, $\Phi \Vdash^{\tau}_{D} \psi$ iff $\Phi \Vdash^{\tau'}_{D} \psi$ and $\tau' \le \tau$. So it is supraclassical. Further properties of the enthymeme consequence relation are the following which are adapted from (Gärdenfors and Makinson 1994).

- Reflexivity: $\Phi \cup \{\alpha\} \Vdash^{\tau}_{D} \alpha$
- Equiv: $\Phi \cup \{\beta\} \Vdash^{\tau}_{D} \gamma$ if $\Phi \cup \{\alpha\} \Vdash^{\tau}_{D} \gamma$ and $\vdash \alpha \leftrightarrow \beta$
- Monotonicity: $\Phi \cup \{\beta\} \Vdash^{\tau}_{D} \alpha$ if $\Phi \Vdash^{\tau}_{D} \alpha$
- Cut: $\Phi \Vdash^{\tau}_{D} \alpha$ if $\Phi \Vdash^{\tau}_{D} \beta$ and $\Phi \cup \{\beta\} \Vdash^{\tau}_{D} \alpha$
- And: $\Phi \Vdash^{\tau}_{D} \alpha \wedge \beta$ if $\Phi \Vdash^{\tau}_{D} \alpha$ and $\Phi \Vdash^{\tau}_{D} \beta$
- Or: $\Phi \cup \{\alpha \vee \beta\} \Vdash^{\tau}_{D} \gamma$ if $\Phi \cup \{\alpha\} \Vdash^{\tau}_{D} \gamma$ and $\Phi \cup \{\beta\} \Vdash^{\tau}_{D} \gamma$

Proposition 9. *The enthymeme inference relation satisfies Reflexivity, Equiv, Cut, Monotonicity, And, and Or.*

Proof. (Reflexivity) If $\alpha \in \Phi$, then $(\emptyset, \emptyset)$ is the solution, and reflexivity holds. (Equiv) Assume $\alpha \equiv \beta$ and $\Phi \cup \{\alpha\} \Vdash^{\tau}_{D} \gamma$. So there is a solution (Γ, S) for $\langle \Phi \cup \{\alpha\}, \gamma \rangle$. So (Γ, S) is a solution for $\langle \Phi \cup \{\beta\}, \gamma \rangle$. So $\Phi \cup \{\beta\} \Vdash^{\tau}_{D} \gamma$. (Monotonicity) Assume $\Phi \Vdash^{\tau}_{D} \alpha$. So there is a solution (Γ, S) for $\langle \Phi, \alpha \rangle$. So there is a solution (Γ', S') for $\langle \Phi \cup \{\beta\}, \alpha \rangle$ where $\Gamma' \subseteq \Gamma$ and $S' \subseteq S$. So $\Phi \Vdash^{\tau}_{D} \beta$. (Cut) Assume $\Phi \Vdash^{\tau}_{D} \beta$ and $\Phi \cup \{\beta\} \Vdash^{\tau}_{D} \alpha$. So there is a solution (Γ_1, S_1) for $\langle \Phi, \beta \rangle$ and there is a solution (Γ_2, S_2) for $\langle \Phi \cup \{\beta\}, \alpha \rangle$. So there is a solution (Γ, S) for $\langle \Phi, \alpha \rangle$ where $\Gamma \subseteq \Gamma_1 \cup \Gamma_2$ and $S \subseteq S_1 \cup S_2$. So $\Phi \Vdash^{\tau}_{D} \alpha$. (And) Assume $\Phi \Vdash^{\tau}_{D} \alpha$ and $\Phi \Vdash^{\tau}_{D} \beta$. So there are solutions (Γ_1, S_2) for $\langle \Phi, \alpha \rangle$ and (Γ_2, S_2) for $\langle \Phi, \beta \rangle$. So there is a solution (Γ, S) for $\langle \Phi, \alpha \wedge \beta \rangle$ where $\Gamma' \subseteq (\Gamma_1 \cup \Gamma_2)$ and $S \subseteq (S_1 \cup S_2)$. So $\Phi \Vdash^{\tau}_{D} \alpha \wedge \beta$. (Or) Assume $\Phi \cup \{\alpha\} \Vdash^{\tau}_{D} \gamma$ and $\Phi \cup \{\beta\} \Vdash^{\tau}_{D} \gamma$. So there are solutions (Γ_1, S_2) for $\langle \Phi \cup \{\alpha\}, \gamma \rangle$. and (Γ_2, S_2) for $\langle \Phi \cup \{\beta\}, \gamma \rangle$. So there is a solution (Γ, S) for $\langle \Phi \cup \{\alpha \vee \beta\}, \gamma \rangle$ where $\Gamma' \subseteq (\Gamma_1 \cup \Gamma_2)$ and $S \subseteq (S_1 \cup S_2)$. So $\Phi \cup \{\alpha \vee \beta\} \Vdash^{\tau}_{D} \gamma$. □

Algorithm 1: The `SUB` algorithm where from atoms $\mathcal{A}$, $\mathsf{Literals}(\mathcal{A})$ is the set of literals, $\mathsf{Clauses}(\mathcal{A})$ is the set of clauses each with 3 literals, and $\mathsf{Subs}(\mathcal{A}, n)$ is the set of substitutions formed with n incoming symbols.

Input: Number of incoming symbols n in a substitution
Output: True if selection is unsatisfiable

$\mathcal{A}$ is a randomly selected set of atoms
β is a randomly selected literal from $\mathsf{Literals}(\mathcal{A})$
Δ is a randomly selected subset from $\mathsf{Clauses}(\mathcal{A})$
S is a randomly selected substitution from $\mathsf{Subs}(\mathcal{A}, n)$
return `UNSAT`$((\Delta \oplus S) \cup \{\neg\beta\})$

So substitutions and abduction of default knowledge are well-integrated with respect to the logical language. Our approach may be viewed as form of abductive reasoning with abstraction axioms (Console and Dupré 1994).

Computing with Substitution and Abduction

We now consider our proposal from a computational point of view. See appendix[1] for code. First, we consider how the cardinality of the substitution, and the proportion of atoms that appear as incoming symbols in the substitution, affect whether or not an inference follows from a set of clauses. To investigate this, we implemented the `SUB` algorithm (Algorithm 1) which randomly selects a set of atoms $\mathcal{A}$, a set of clauses Δ, a query β, and a substitution S, and then calls `UNSAT`$((\Delta \oplus S) \cup \{\neg\beta\})$ which is a call to a SAT solver. If the algorithm returns `TRUE`, the set $(\Delta \oplus S) \cup \{\neg\beta\}$ is inconsistent and hence the query follows from $\Delta \oplus S$, whereas if it returns `FALSE`, the set $(\Delta \oplus S) \cup \{\neg\beta\}$ is consistent and hence the query does not follow from $\Delta \oplus S$. The `SUB` algorithm is coded in Python, and uses the PySAT implementation (Ignatiev, Morgado, and Marques-Silva 2018) that incorporates a range of SAT solvers (e.g. Glucose3).

Using Algorithm 1, we undertook an empirical investigation on the number of substitutions required to influence the inferences that follow from a knowledgebase. The results are reported in Figure 1. The consistency ratio is the proportion of runs that are consistent (i.e. the inference does not follow). So we see that increasing the cardinality of the substitution (i.e. the number of swaps in the substitution), the number of inferences rises. Each line in Figure 1 denotes the number of atoms in the language. Since, the number of incoming atoms is fixed, decreasing the number of atoms means that the ratio of incoming atoms to all atoms rises. So for each line in Figure 1 going from top to bottom, the consistency ratio falls. In other words, increasing the ratio

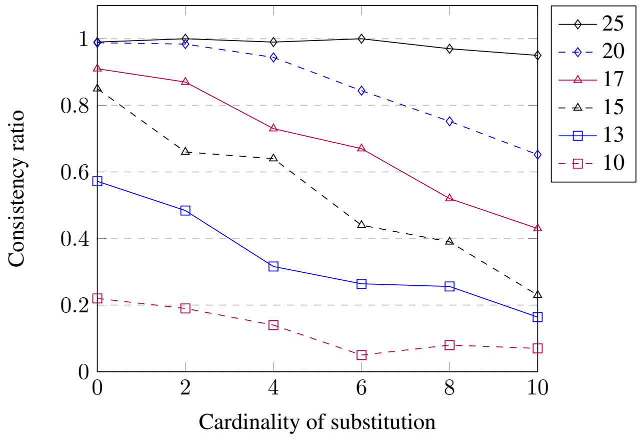


Figure 1: Experiments with the SUB algorithm (Algorithm 1). The knowledgebase size is 50 clauses with each substitution having 10 incoming symbols. Each line is for a specific number of atoms in the language as defined in the legend (top right corner). Consistency ratio is the proportion out of 100 runs where the selection is consistent.

of incoming atoms to all atoms, means the number of inferences rises. These results show how substitution size, and ratio of incoming atoms to all atoms, can effect the number of inferences we get from a knowledgebase.

Next we consider a simple algorithm to compute abductive solutions (Algorithm 2). It considers smaller sets of defaults and substitutions before considering larger sets thereby ensuring minimality. Every pair (Γ, S) returned by the algorithm is an abductive solution, and if there is an abductive solution, then the algorithm will find one.

Proposition 10. *Let Δ and Π be sets of clauses and β be a literal. (1) If* $\texttt{ABDUCE}(\Delta, \Pi, \beta, \mathcal{S})$ *returns* (Γ, S)*, then* (Γ, S) *is an abductive solution of* (Δ, β)*. (2) If there is an abductive solution of* (Δ, β)*, then there is a* (Γ, S) *such that* $\texttt{ABDUCE}(\Delta, \Pi, \beta, \mathcal{S})$ *returns* (Γ, S)*.*

Proof. (1) Assume $\texttt{ABDUCE}(\Delta, \Pi, \beta, \mathcal{S})$ returns (Γ, S). So there is a $S \subseteq \mathcal{S}$ and $\Gamma \subseteq \Pi$ s.t. $\texttt{UNSAT}(((\Delta \cup \Gamma) \oplus S) \cup \{\neg\beta\})$, i.e. the entailment condition, and there is no $S' \subset S$ s.t. $\texttt{UNSAT}(((\Delta \cup \Gamma) \oplus S') \cup \{\neg\beta\})$, i.e. the minimum swaps condition, and there is no $\Gamma' \subset \Gamma$ s.t. $\texttt{UNSAT}(((\Delta \cup \Gamma') \oplus S) \cup \{\neg\beta\})$, i.e. the minimum assumptions condition. Hence, (Γ, S) is an abductive solution. (2) Follows similarly. □

As an indication of the performance, for a knowledgebase of 100 clauses, a set of 10 defaults, and a substitution of 10 swaps, with 5 outing symbols, randomly generated from 20 atoms, the ABDUCE algorithm takes 0.22 seconds (average of 50 runs) on a Windows 10 laptop (AMD A10 Radeon R8 with 8GB RAM). Since the implementation naively considers each subset of substitutions and defaults for finding abductive solutions, it does not scale well. However, if we can accept approximate abductive solutions (i.e. a set of defaults and a substitution that is sufficient for deriving the query, but is not necessarily minimal), then we can scale the algorithm substantially (e.g. we can identify a non-minimal solution for a knowledgebase of 1000 clause, 1000 defaults, a substitution of 100, with 100 atoms, of which 50 are incoming symbols, in 0.24 secs, as an average of 50 runs).

Algorithm 2: The ABDUCE algorithm where $|.|$ returns the cardinality of the set.

Input: Δ and Π are sets of clauses, β is a literal, and $\mathcal{S}$ is a set of substitutions.
Output: Abductive solution (Γ, S).

```
for n = 0 to |S| do
  for m = 0 to |Π| do
    for next (Γ, S) s.t. |S| = n and |Γ| = m do
      if UNSAT(((Δ ∪ Γ) ⊕ S) ∪ {¬β}) then
        return (Γ, S)
return No abductive solution
```

Discussion

The proposal in this paper can be used as part of a solution to bridge argument mining and logic-based argumentation. Argument mining can identify premises and claims within text, whereas logic-based argumentation can be used to build arguments from logical formulae so that the premises imply the claim, and this can be constructed or checked algorithmically. To bridge these two approaches, we can extract premises and claims from text using argument mining, and then represent the premises and claims by logical formulae. But as arguments in text are enthymemes, there will be logical formulae missing. With our proposal, we can look for appropriate commonsense knowledge and substitutions, and then automatically check consistency or validity of the logic-based arguments. Decodings might be incorrect, but this may be tolerable for example when a dialogue allows for the participants to ask questions and offer clarifications (e.g. (Xydis et al. 2020)). The decoded enthymemes can then be used for tasks such as updating an argument graph (e.g. (Mailly 2016)), fusion (e.g. (Santini, Jøsang, and Pini 2018)), or explanations (e.g. (Becker et al. 2020)).

In future work, we will investigate extending the approach to other approaches to structured argument including ASPIC+ (Prakken 2010), ABA (Toni 2014), and defeasible logic programming (García and Simari 2004), we will investigate more efficient algorithms (e.g. based on (Koitz-Hristov and Wotawa 2020)), we will quantify the uncertainty of a decoding being correct, and then use this in probabilistic argumentation, we will investigate harnessing richer non-monotonic formalisms for representing commonsense knowledge (e.g. (Brewka 1991; Mueller 2006; Davis 2017; D'Asaro et al. 2020; Vassiliades et al. 2020)), and we will investigate how we can harness richer language models for contextualized word embeddings (e.g. Elmo (Peters et al. 2018) and BERT (Chang et al. 2019)) in order to identify, represent, and reason with, context-sensitive substitutions.

References

Atkinson, K.; Baroni, P.; Giacomin, M.; Hunter, A.; Prakken, H.; Reed, C.; Simari, G.; Thimm, M.; and Villata,

S. 2017. Towards Artificial Argumentation. *AI Magazine*, 38(3): 25–36.

Becker, M.; Hulpus, I.; Opitz, J.; Paul, D.; Kobbe, J.; Stuckenschmidt, H.; and Frank, A. 2020. Explaining arguments with background knowledge. *Datenbank-Spektrum*, 20(2): 131–141.

Besnard, P.; and Hunter, A. 2001. A logic-based theory of deductive arguments. *Artificial Intelligence*, 128: 203–235.

Black, E.; and Hunter, A. 2012. A relevance-theoretic framework for constructing and deconstructing enthymemes. *J. Logic and Computation*, 22(1): 55–78.

Brewka, G. 1991. *Nonmonotonic Reasoning: Logical Foundations of Commonsense*. Cambridge University Press.

Budanitsky, A.; and Hirst, G. 2006. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1): 13–47.

Chang, M.; Toutanova, K.; Lee, K.; and Devlin, J. 2019. Language Model Pre-training for Hierarchical Document Representations. *CoRR*, abs/1901.09128.

Console, L.; and Dupré, D. 1994. Abductive reasoning with abstraction axioms. In *Foundations of Knowledge Representation and Reasoning*, volume 810, 98–112. Springer.

D'Asaro, F. A.; Bikakis, A.; Dickens, L.; and Miller, R. 2020. Probabilistic reasoning about epistemic action narratives. *Artificial Intelligence*, 287: 103352.

Davis, E. 2017. Logical Formalizations of Commonsense Reasoning: A Survey. *Journal of Artificial Intellegence Research*, 59: 651–723.

Dung, P. 1995. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2): 321–358.

García, A.; and Simari, G. 2004. Defeasible logic programming: an argumentative approach. *Theory and Practice of Logic Programming*, 4(1-2): 95–138.

Gärdenfors, P.; and Makinson, D. 1994. Nonmonotonic Inference Based on Expectations. *Artificial Intelligence*, 65(2): 197–245.

Gorogiannis, N.; and Hunter, A. 2011. Instantiating abstract argumentation with classical logic arguments: Postulates and properties. *Artificial Intelligence*, 175(9-10): 1479–1497.

Hosseini, S.; Modgil, S.; and Rodrigues, O. 2014. Enthymeme construction in dialogues using shared knowledge. In *Proc. of COMMA'14*, volume 266 of *FAIA*, 325–332. IOS Press.

Hu, B.; Kalfoglou, Y.; Alani, H.; Dupplaw, D.; Lewis, P.; and Shadbolt, N. 2006. Semantic Metrics. In *Proc. of EKAW'06*, volume 4248 of *LNCS*, 166–181. Springer.

Hunter, A. 2007. Real arguments are approximate arguments. In *Proc. of AAAI'07*, 66–71. AAAI Press.

Ignatiev, A.; Morgado, A.; and Marques-Silva, J. 2018. PySAT: A Python Toolkit for Prototyping with SAT Oracles. In *Proc. of SAT'18*, volume 10929 of *LNCS*, 428–437. Springer.

Koitz-Hristov, R.; and Wotawa, F. 2020. Faster horn diagnosis - a performance comparison of abductive reasoning algorithms. *Applied Intelligence*, 50(5): 1558–1572.

Mailly, J. 2016. Using enthymemes to fill the gap between logical argumentation and revision of abstract argumentation frameworks. *CoRR*, 1603.08789.

Mendes, P.; Jakob, M.; García-Silva, A.; and Bizer, C. 2011. DBpedia spotlight: shedding light on the web of documents. In *Proc. of I-SEMANTICS'11*, 1–8. ACM.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *CoRR*, 1301.3781.

Miller, G. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11): 39–41.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *Proc. of ACL'09*, 1003–1011. Association for Computer Linguistics.

Mueller, E. T. 2006. *Commonsense Reasoning*. Morgan Kaufmann. ISBN 0-12-369388-8.

Paulheim, H. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 8(3): 489–508.

Pennington, J.; Socher, R.; and Manning, C. 2014. GloVe: Global vectors for word representation. In *Proceedings of (EMNLP)'14*, 1532–1543.

Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep Contextualized Word Representations. In *Proc. of NAACL-HLT'18*, 2227–2237. CL.

Petroni, F. 2010. Measures of lexical distance between languages. *Physica A: Statistical Mechanics and its Applications*, 11(1): 2280–2283.

Prakken, H. 2010. An abstract framework for argumentation with structured arguments. *Argument and Computation*, 1(2): 93–124.

Rector, A.; and Brandt, S. 2008. Viewpoint paper: Why do it the hard way? The case for an expressive description logic for SNOMED. *J. American Medical Informatics Association*, 15(6): 744–751.

Santini, F.; Jøsang, A.; and Pini, M. 2018. Are my arguments trustworthy? Abstract argumentation with subjective logic. In *Proc. of FUSION'18*, 1982–1989. IEEE.

Toni, F. 2014. A tutorial on assumption-based argumentation. *Argument and Computation*, 5(1): 89–117.

Vassiliades, A.; Patkos, T.; Bikakis, A.; Flouris, G.; Bassiliades, N.; and Plexousakis, D. 2020. Preliminary notions of arguments from commonsense knowledge. In *Proc. of SETN'20*, 211–214. ACM.

Walton, D. 2001. Enthymemes, common knowledge and plausible inference. *Philosophy and Rhetoric*, 34(2): 93–112.

Xydis, A.; Hampson, C.; Modgil, S.; and Black, E. 2020. Enthymemes in dialogues. In *Proc. of COMMA'20*, volume 326 of *FAIA*, 395–402. IOS Press.