

創作物におけるライセンス情報の構造化と それに基づく互換性評価システム

町田 渉¹ 梶 克彦¹

概要：本研究では、創作物におけるライセンス情報の構造化と、ドメインをまたぐ互換性評価を実現するシステム「LicenSeed」を提案する。既存のライセンス管理はソフトウェア分野に特化しており、多様な創作活動への適用や柔軟なカスタマイズが困難であった。そこで本研究では、ライセンス情報を構造化した「LicenSeed Spec」と、WordNet および fastText を用いた意味的類似度に基づく互換性評価モデルを構築した。これにより、クリエイターは表現を損なわずにライセンスを提示でき、利用者は異なるドメイン間でもライセンスの互換性を定量的に把握できる。

1. はじめに

ソフトウェア分野では MIT, Apache, GPL など明文化されたライセンスが普及し、choosealicense などのツールにより適切なライセンス選択が支援されている。OSS コミュニティではライセンス管理体系が確立されているといえる。

一方で、既存のライセンス管理手法は特定ドメインに限定されており、カスタマイズ性に乏しいという課題がある。OSS 文脈の SPDX や choosealicense はソフトウェアに特化しており、イラスト・音楽等への適用が困難である。Creative Commons の「コモンズ証」は視覚的理解を促進するものの、カスタムな条件を含む事例には適用しづらい。このように、特定のドメインに依存しない柔軟な記述と構造化の手法が不足している。

創作活動への拡張には用語体系の未整備やライセンス情報の散逸という問題がある。ソフトウェア特化の用語体系(派生物、ソースコード開示など)が創作物に馴染まない。文化庁調査 [1] によると、利用者の約 85% が著作権について正確に知っている自信がなく、クリエイターの約 40% が制作物に対して著作権表示をしていないという実態が明らかになった。また、独自のライセンス文が散逸・長文化・専門用語化しており、読み飛ばされやすい。

本研究は、特定のドメインに限定せず「ライセンスの構造化」と「ドメインをまたぐ互換性評価」を実現するシステム「LicenSeed」を提案する。LicenSeed は、ライセンス情報の構造化仕様である「LicenSeed Spec」と、それを扱

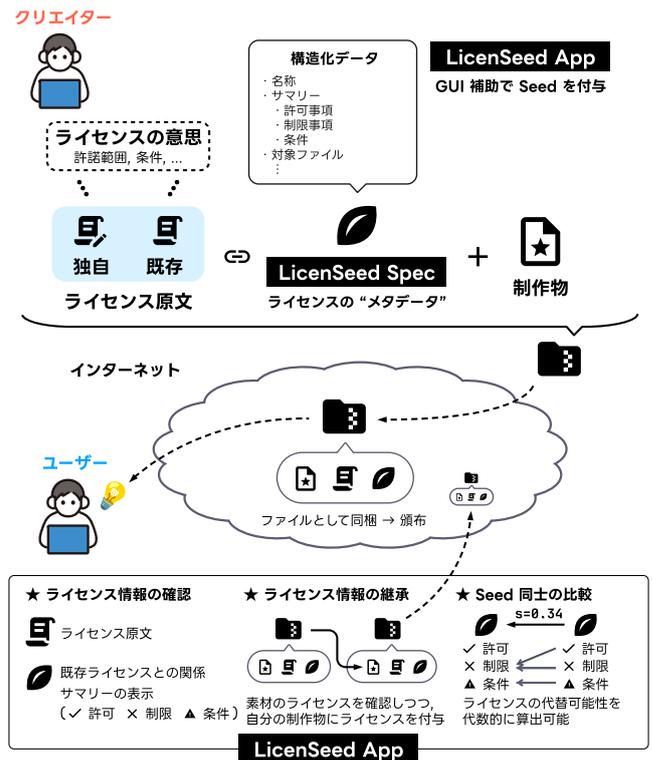


図 1 提案システムの全体像

うアプリケーション「LicenSeed App」から構成される。

構造化へのアプローチとして、クリエイターが表現を損なわずにライセンスを提示できる枠組みを提供する。許可・制限・条件を一目で把握できるサマリーとライセンス文の整合性を保証し、ライセンス情報を YAML で記述した構造化データ「シード」およびその規格である「LicenSeed Spec」を提案する。

¹ 愛知工業大学 情報科学部

互換性評価へのアプローチとして、単なる可否判定（0か1か）ではなく、互換性をスコア化する代数的な評価を実現する。定義済み用語間の意味の類似度（WordNet と fastText）を統合し、妥協点や推薦の根拠を示す。

2. 関連研究

2.1 ライセンス分類と互換性モデル

CaLi (Compatibility and Compliance Lattice) モデル [2], [3] はライセンス間の互換性を格子モデルで表現する数学的フレームワークである。 $l_i \rightarrow l_j$ を許可 (P), 禁止 (Pr), 義務 (O) の包含関係で定義し、ライセンス間の互換性を数学的に表現する。具体例として CC_CaLi は Creative Commons に特化した評価指標を提示している。CC ライセンスの組み合わせに対する互換性スコアを算出し、制限の格子構造を活用した互換性判定手法を提案する。

しかし既存研究には以下の限界がある。第 1 に、単一のライセンス群内での評価（互換・非互換の二値判定）に留まっており、ドメインをまたぐ横断的な評価ができない。第 2 に、妥協点の探索（代数的な評価）ができないため、「完全一致ではないが許容範囲」といった柔軟な判断を支援できない。

2.2 既存のライセンス管理システム

SPDX [4] と choosealicense [5] はソフトウェア分野で広く利用されているが、創作物への適用には限界がある。SPDX は主に OSS ライブラリの依存関係管理に利用され、創作物全般への説明粒度が不足している。

Creative Commons [6] はコモンズ証により視覚的な理解を促進するが、カスタムな条件への対応が困難である。その他、VN3 ライセンス [7] などの VRChat 等の特定プラットフォーム内でのライセンスシステムも存在するが、汎用性に課題がある。既存システムはいずれもライセンス文とサマリーの整合性の保証が不十分である。なお、権利表現言語として W3C の ODRL [8] 等が存在するが、XML/JSON-LD ベースであり一般のクリエイターには、またこれも記述が複雑である。

3. LicenSeed Spec の設計

3.1 要求仕様と構造化

提案システムに求められる要求仕様を以下に整理する。第 1 に、ドメインをまたぐ互換性を定量的に算出するため、ライセンス条件を機械可読な形式で構造化する必要がある。第 2 に、「商用利用だけ NG」といった微妙な差異を比較可能にするため、許可・制限・条件を明確に分離したデータ構造が必要である。第 3 に、クリエイターが表現を損なわずにライセンスを提示できるよう、サマリーと全文の整合性を保証する仕組みが必要である。

これらの要求仕様を満たすため、許可・制限・条件を明

確に分離したデータ構造を採用した。

本研究では、ライセンス情報を構造化するためのデータ構造およびファイル形式の仕様として「LicenSeed Spec」を定義する。LicenSeed Spec に基づいて記述された個別のライセンスデータファイルを「シード」と呼ぶ。シードは YAML 形式で記述され、サマリー・変数・本文・ハッシュを一体化する。シードは 4 種類に体系化される。

ベースシードは代表的なライセンスを登録し、グループごとに共通用語を定義する。再利用シードは変数置換のみで適用可能であり、最も簡便な利用形態である。フォークシードは特記事項を追加し、既存ライセンスを部分的に調整する。カスタムシードは Markdown を活用して独自条件を記述し、構造化と柔軟性を両立する。定義済み用語 (TERM) と Markdown の併用により、ケースバイケースの条件を構造的に表現できる。

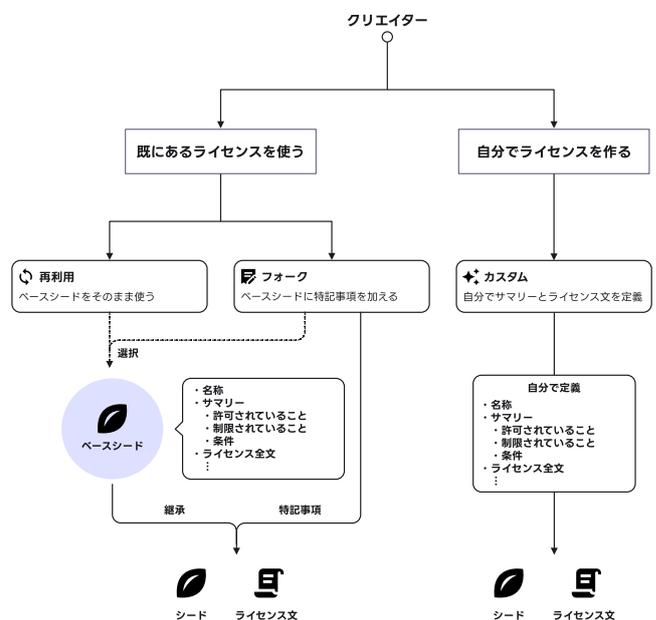


図 2 シードの種類とデータ構造

3.2 互換性算出

互換性算出 (Calculation) は、ライセンスが代替可能かをスコア化するプロセスである。CaLi を参考にしつつ、ドメインをまたぐ評価と代数的な評価を実現するために独自のスコア化を行う。

互換性算出では、許可・制限・条件の定義済み用語集合 A_p, A_l, A_c と B_p, B_l, B_c を比較してスコアを算出する。本手法の特徴は、「A を B で代替できるか」という非対称的な評価を実現する点にある。制限を最重視 (重み 3) し、続けて、許可 (重み 2), 条件 (重み 1) として加重和を計算する。

許可 (Permissions) のスコアは、両ライセンスに共通する許可項目を重視し、不足項目を減点する方針で以下のよ

うに定義する。

$$p_{A \leftarrow B} = \frac{|A_p \cap B_p| - |A_p \setminus B_p|}{|A_p|} \quad (1)$$

ここで、 $A_p \cap B_p$ は両方に含まれる許可（互換性を高める）、 $A_p \setminus B_p$ は A に必要だが B にはない許可（互換性を下げる）を表す。 $|A_p| = 0$ のときは $p_{A \leftarrow B} = 1$ とし、許可要求がない場合は互換とみなす。

制限（Limitations）のスコアは、追加の制限を厳しく減点し、整合性を最重視する方針で以下のように定義する。

$$l_{A \leftarrow B} = \frac{|A_l \cap B_l| - |B_l \setminus A_l|}{|A_l|} \quad (2)$$

ここで、 $A_l \cap B_l$ は両方に含まれる制限（整合性を高める）、 $B_l \setminus A_l$ は B に追加の制限（互換性を下げる）を表す。ただし、境界条件は、

$$l_{A \leftarrow B} = \begin{cases} 1 & (|A_l| = 0 \wedge |B_l| = 0) \\ -|B_l| & (|A_l| = 0 \wedge |B_l| \neq 0) \end{cases} \quad (3)$$

を用いる。

条件（Conditions）のスコアは制限と同様の計算方法を用い、条件についても境界条件を適用する。

$$c_{A \leftarrow B} = \frac{|A_c \cap B_c| - |B_c \setminus A_c|}{|A_c|} \quad (4)$$

最終的な互換性スコアは、以下の加重和として算出する。

$$S_{A \leftarrow B} = 2 \cdot p_{A \leftarrow B} + 3 \cdot l_{A \leftarrow B} + 1 \cdot c_{A \leftarrow B} \quad (5)$$

互換性評価において、CaLi モデルのような格子構造による厳密な順序関係だけでは、ドメインをまたぐ際の柔軟な「妥協点」を見出しにくい。また、ライセンス A を B で代替する際の評価は非対称であるべきである。そこで本手法では、各用語に対してアクション状態を定義し、状態間の遷移にスコアを与えることで、権利の拡大・縮小を定量的に評価する。アクション状態として、Undefined（明示的な言及なし）、Permission（許可）、Condition（条件付き許可）、Limitation（禁止）の 4 状態を定義する。遷移ペナルティの設定では、制限状態が緩和される場合（Limitation → Permission/Undefined）は加点、逆方向の遷移（Permission → Limitation）は減点として扱う。具体的なペナルティ値を表 1 に示す。

表 1 遷移ペナルティ行列

	Undefined	Permission	Condition	Limitation
Undefined	0.0	+0.3	+0.1	-0.5
Permission	-0.3	0.0	-0.2	-1.0
Condition	-0.1	+0.2	0.0	-0.8
Limitation	+0.5	+1.0	+0.5	0.0

異なるシードグループ間の互換性評価では、定義済み用

語の意味的類似度を考慮する必要がある。類似度行列の生成には、WordNet [9] と fastText [10] を併用する。まず WordNet を用いて同義語や上位・下位関係に基づく類似性を評価し、WordNet でカバーされない用語や関係性については fastText の分散表現を用いた cos 類似度で補完する。これらを統合し、用語間の類似度スコア (0.0~1.0) を持つ行列を生成する。互換性スコアの計算においては、予備検討に基づき閾値 $\theta = 0.7$ を設定する。交集合への寄与として用語間の最大類似度スコアをそのまま加算し（完全一致は 1.0）、差集合への寄与として用語間の最大類似度が閾値 θ 未満の場合は異なる用語とみなして 1.0 を加算する。一方、閾値以上の場合は実質的に同一の用語とみなし、差集合には加算しない (0.0)。

4. LicenSeed App の実装

本章では、前章で設計した LicenSeed Spec に基づくライセンス管理アプリケーション「LicenSeed App」の実装について述べる。本アプリケーションは Electron および React を用いて実装され、クロスプラットフォームで動作する。主な機能として、シードの作成、表示、比較を提供する。

4.1 シードの作成



図 3 ベースシードの選択画面

シード作成機能は、クリエイターが自身の作品に適したライセンスを選択・調整し、配布用のファイルを生成する機能である。UI はウィザード形式を採用しており、ユーザーは以下の手順で操作を行う。まず、ベースシードの選択を行う（図 3）。@software/MIT や @commons/CC-BY などのプリセットから選択すると、ライセンス文とサマリーのプレビューが表示される。次に、変数入力を行う。著作権者名、年、連絡先などを入力すると、リアルタイムプレビューによりライセンス文中の変数が即座に置換される。最後に、必要に応じて特記事項を追加する。Markdown エディタを用いて独自の条件を記述でき、定義済み用語を併用することで構造化データを維持しつつ柔軟な表現が可能となる。

生成プロセスでは、LicenSeed Spec に準拠したシードファイル (LICENSEED.yml) と、一般的なライセンスファイル (LICENSE) の2つが出力される。この際、システムは生成されたライセンス本文の正規化を行い、SHA-256 ハッシュ値を計算する。このハッシュ値はシードファイル内にメタデータとして埋め込まれる。

4.2 シードの表示

シード表示機能は、利用者が作品に付与されたライセンス条件を正確かつ直感的に把握するための表示機能である。ユーザーが作品に同梱された LICENSEED.yml を読み込むと、アプリケーションは YAML データをパースし、許可・制限・条件をセクションごとに整理して表示する。各項目はアイコンと色分け（許可は青、制限は赤など）によって視覚化され、専門知識がない利用者でも権利関係を一目で把握できるよう設計した。また、Markdown で記述された特記事項は HTML としてレンダリングされ、リンクや強調表現を含むリッチなテキストとして表示される。

本機能の重要な役割として、ライセンスの整合性検証がある。シード読み込み時に、同梱されている LICENSE ファイルのハッシュ値を再計算し、シードファイルに記録されたハッシュ値と照合する。不一致の場合、サマリーは表示されない。これにより、LICENSE ファイルが常に正の扱いになる。

4.3 シードの比較

シード比較機能は、複数のライセンス間の互換性や差異を定量的に評価する機能である。ユーザーが比較対象となる2つのシードを選択すると、システムは3.2で述べた互換性算出ロジックに基づいてスコア計算を行う。UI上では、両者の許可・制限・条件が対比形式で表示され、差異がある項目がハイライトされる。また、定義済み用語間の意味的類似度 (WordNet および fastText) を用いたマッチングにより、表記が異なる場合でも意味的に近い条件を関連付けて提示する。これにより、利用者は「商用利用が可能か」「クレジット表記が必要か」といった実質的な条件の違いを、ドメインや用語の違いを超えて理解することが可能となる。

5. 動作テスト

5.1 ベースシードの収集

まず既存の OSS ライセンスや Creative Commons ライセンスをシード形式に変換・収集した。

@software グループは choosealicense から全ての OSS ライセンスを抽出・変換した。MIT, Apache-2.0, GPL-3.0, BSD-3-Clause などが含まれる。

@commons グループは Creative Commons ライセンスを同様に抽出・変換した。CC0-1.0, CC-BY-4.0, CC-BY-SA-

4.0 などが含まれる。

5.2 互換性算出の妥当性

提案手法における互換性算出の妥当性を検証するため、体系化されたライセンス群である Creative Commons (CC) ライセンスを用いて評価を行った。比較対象として、ライセンスの互換性を格子モデルで表現する既存研究 CC_CaLi [2] のグラフ構造を採用した。CC_CaLi では、許可 (P), 禁止 (Pr), 義務 (O) の包含関係に基づき、ライセンス間の親子関係 (矢印) が定義されている。

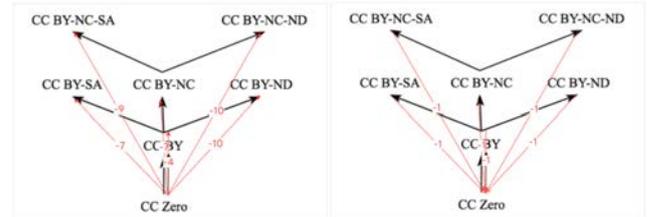


図4 CC ライセンス間の互換性スコアと CC_CaLi グラフの比較

提案手法に基づき CC ライセンス間の互換性スコアを算出し、その結果を CC_CaLi のグラフ上に重ね合わせたものを図4に示す。図中の赤線および数値は、LicenSeed Spec 同士の比較によって導出された互換性スコアを表している。検証の結果、提案手法による算出結果は CC_CaLi が示すライセンスの包含関係と構造的に一致することが確認された。具体的には、制約の緩いライセンス (CC0 や CC BY) から、より制約の厳しいライセンス (CC BY-NC など) への派生関係において、スコアが有向グラフの辺 (エッジ) として正しく検出されている。これにより、本提案手法である代数的なスコア算出が、既存の厳密な格子モデルによる互換性判定と同等の妥当性を持ち、かつその関係を定量的に表現できているといえる。

5.3 シナリオに基づく動作検証

本システムの実用性を検証するため、想定される3つの利用シナリオ (クリエイターによる付与, 利用者による確認, 二次創作時の利用) に基づき、LicenSeed App 上で動作テストを行った。

5.3.1 クリエイターによるライセンス付与

ペルソナ A (クリエイター) は、自作のマウスカーソル素材 (17ファイル) に対して「CC-BY-NC (表示 - 非営利)」を付与し、かつ独自のクレジット表記形式を指定することを想定した。

図5に示すシード選択画面において、クリエイターはウィザード形式に従い操作を行う。まず、図3のようにベースシードとして @general グループから CC-BY-NC-4.0 を選択する。次に、適用範囲 (Territory) として assets/* を指定することで、ディレクトリ内の全ファイルへ一括して



図 5 シードの付与画面

ライセンスを適用できることを確認した。さらに、特記事項 (Notes) としてクレジット表記の具体的なフォーマットを Markdown 形式で記述し、これらがリアルタイムプレビューに反映されることを確認した。

5.3.2 利用者によるライセンス確認

ペルソナ B (利用者) は、配布された制作物のライセンス範囲を把握し、クレジット表記を行うことを想定した。



図 6 シードの閲覧画面

利用者は制作物に同梱された LICENSEED.yml を本システムで開くことで、図 6 に示す閲覧画面にアクセスする。検証の結果、以下の UX が実現されていることを確認した。第 1 に、ライセンスの専門知識がない利用者でも、「許可 (Permissions)」「制限 (Limitations)」「条件 (Conditions)」がアイコンと色分けにより一目で識別可能であった。第 2 に、シナリオ A で追加されたクレジット表記の特記事項がレンダリングされ、利用者が容易にコピー&ペーストして利用できることが確認された。第 3 に、必要に応じて「ライセンス全文」タブへ切り替えることで、法的な原文 (LICENSE ファイルの内容) へシームレスにアクセス可能であった。

5.3.3 二次創作時の互換性確認

ペルソナ C は、複数の素材 (素材 A と素材 B) を組み合



図 7 シードの閲覧画面 (継承含む)

わせて新しい作品を制作する状況を想定した。

本システムでは、複数のシードファイル (LICENSEED.yml) を同一プロジェクト内で読み込むことが可能である。動作テストでは、異なるライセンスが付与された複数の素材をインポートし、それぞれの許諾範囲や制限事項を並列して確認できることを検証した。これにより、クリエイターは素材間のライセンスの衝突 (例えば、商用利用の可否や継承条件の不一致など) を制作前に予見することが容易となる。

5.3.4 ライセンス差分の確認

ペルソナ C は、利用を検討している 2 つのライセンス (Open Software License 3.0 と Vim License) の差異を詳細に比較する状況を想定した。



図 8 シードの比較画面

本システムでは、2 つのシードファイル (LICENSEED.yml) を比較し、それぞれの許諾範囲や制限事項の差異を確認することが可能である。動作テストでは、図 8 に示すように、異なるライセンス間でも定義済み用語の意味的類似度に基づいて対応関係が可視化されることを検証した。例えば、OSL-3.0 の「特許利用」や「ネットワーク利用は配布」といった独自の条件が、Vim License には存在しないことや、商標利用の制限などの差異が視覚的に明示された。これにより、クリエイターはライセンス間の詳細な条件の違いを制作前に予見し、適切なライセンス選定を行うことが容易となる。

5.4 生成物の整合性確認

動作テストの最終段階として、生成されたファイルの整合性を確認した。図9は、本システムによって出力されたLICENSEファイル（原文）およびLICENSEED.yml（シード）である。ライセンス原文はベースシードに基づき正確に生成されており、シードファイルにはメタデータとして適用範囲や特記事項が構造化されて保存されている。また、生成時に算出されたハッシュ値により、両ファイルの整合性が担保されていることを確認した。

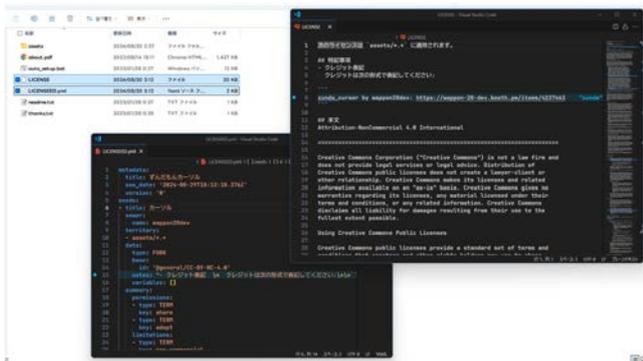


図9 生成されたLICENSEファイルとLICENSEED.yml

6. おわりに

本研究では、特定のドメインに限定されないライセンス管理システムLicenSeedを提案した。LicenSeed Specによるライセンス情報の構造化と、意味的類似度に基づく互換性評価により、ドメインをまたぐ柔軟なライセンス管理を実現した。互換性算出では、既存のCC_CaLiスコアとの整合性が確認され、提案手法の妥当性が示された。

今後の課題として、制作物の種類も考慮したライセンス情報による検索・提案システムの構築が挙げられる。また、制作物とLicenSeed Specおよびライセンス原文が結びついたプラットフォームの開発を目指す。ライセンスの継承関係や依存関係を可視化することで、制作者への感謝の循環を促進する仕組みを検討する。

参考文献

- [1] 文化庁, 株式会社クロス・マーケティング. 令和4年度著作権に関するアンケート調査, 2023. Accessed: 2024-08-16.
- [2] Benjamin Moreau, Patricia Serrano-Alvarado, and Emmanuel Desmontils. CaLi: A lattice-based model for license classifications. In *34ème Conférence sur la Gestion de Données – Principes, Technologies et Applications (BDA 2018)*, Bucharest, Romania, October 2018. hal-01934596.
- [3] Benjamin Moreau, Patricia Serrano-Alvarado, Matthieu Perrin, and Emmanuel Desmontils. *Modelling the Compatibility of Licenses*, pp. 255–269. Springer, 05 2019.
- [4] The Linux Foundation. The Software Package Data Exchange (SPDX) Specification Version 2.3. <https://spdx.github.io/spdx-spec/v2.3/>, 2024. Accessed:

2025-12-22.

- [5] GitHub, Inc. Choose a License. <https://choosealicense.com/>, 2024. Accessed: 2024-05-07.
- [6] Creative Commons. Creative Commons. <https://creativecommons.org/>, 2024. Accessed: 2025-06-23.
- [7] あしやまひろこ, en.129, ろーてく, ほか. VN3 License. <https://www.vn3.org/>, 2024. Accessed: 2024-05-14.
- [8] Renato Iannella and Michael Steer. OdrI information model 2.2. Technical report, W3C Recommendation, 2018.
- [9] George A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, Vol. 38, No. 11, p. 39–41, November 1995.
- [10] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.